DOCUMENT RESUME

ED 180 441 IR 007 915

AUTHOR	Stentz, Michael, Ed.; Motsinger, Linda, Ed.
TITLE	Proceedings of the Annual Conference on Academic Computing Applications (6th, Indiana University Northwest, Bloomington, Indiana, April 6, 1979).
INSTITUTION	Indiana Univ. Northwest, Gary.
PUB DATE	6 Apr 79
NOTE	237p.; Some examples and figures may not reproduce
AVAILABLE FROM	Wrubel Computing Center, Indiana University, Mem W 002, Bloomington, IN 47401 ($3.50 per copy)
EDRS PRICE	MF01/PC10 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Oriented Programs; Computer Programs; Data Analysis; Data Collection; *Educational Research; *Information Systems; Music Education; *Networks; Simulation

ABSTRACT

Topics which range from the more popular computing applications in accounting, statistics, and administration to the less ordinary applications of the computer to the fields of fine arts, medicine, and linguistics, are discussed in this collection of 22 conference papers. The papers are divided into four tracks: the first deals with statistical computing and methods for teaching it; the second is concerned with applications of the computer in theater and journalism, music, the arts, and general utility programs; the third includes information about administration, computer simulations, plotting applications, and computing futures; and a concluding paper summarizes a grammatical concordance package program for the Greek New Testament. (JEG)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED180441

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

THE INDIANA UNIVERSITY

COMPUTING NETWORK

BEST COPY AVAILABLE

6th ANNUAL CONFERENCE

ON

ACADEMIC COMPUTING APPLICATIONS

INDIANA UNIVERSITY NORTHWEST

April 6, 1979

Director:

Robert Andree
Indiana University Northwest

Proceedings Editors:

Michael Stentz
Linda Motsinger

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Paul L. Glenn

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR007915

INDIANA UNIVERSITY
Office of Information and Computer Services
BRYAN HALL 7
BLOOMINGTON, INDIANA 47401

April 6, 1979

TEL. NO. 812—337-1053

Greetings:

The annual Academic Computing Applications Conference has become a traditional event for Indiana University. We are especially pleased that it represents the concept of a University system by being rotated from one campus to another. This year we are grateful to Indiana University Northwest for hosting the conference. This regional campus has a rich heritage in academic computing that continues to grow.

This sixth annual event has special significance for several persons. Six years ago the members of the Computer Network Operations Advisory Committee originated this conference. It has been through the dedicated work of members of this committee that the arrangements are made for this conference. For several key persons this represents their last conference in their current role. Gordon Wakefield, Assistant Director at IUPU Fort Wayne; Michael Stentz, Regional Campus Coordinator at Wrubel Computing Center; Robert Wayman, Coordinator of Administrative User Services at I. U. Southeast; and Lloyd A. Case, Director of Computer Services at I. U. Southeast have all accepted new positions outside Indiana University. To them we give our sincere thanks for the meaningful contributions they made to the Indiana University Computing Network and this conference in particular.

Although she will not be leaving Indiana University, Kristin Froehlke, Director of I. U. South Bend, has now become Associate Director of Computing Services for Technical Services at IUPUI. As a consequence she relinquishes the chairmanship of the Computer Network Operations Advisory Committee, a post she has held since its founding in 1971. A special thanks to you Kris.

To the participants from throughout the Indiana University community and from our sister institutions, our personal thanks for your active involvement and continued pursuit of academic excellence through the use of computing resources.

Sincerely,

Thomas W. West

Thomas W. West,
University Director of
Information and Computer Services

ACKNOWLEDGMENTS

I wish to acknowledge the willing support of the administration at I. U. Northwest in providing the facilities, the staff of the computer center for providing the effort necessary to get the conference started, and the regional campus directors for the effort and assistance necessary to make the conference so.

Thanks go to Linda Motsinger, Paul Glenn, and Bruce Ashby for their efforts in assembling the proceedings, publications, and mailings. Lastly a special note of thanks goes to Mickey Stentz. Mickey has been the driving force who has organized, publicized, and turned over to the conference director the last three conferences in time for the conference director to take the bow. I am thankful that I was able to take advantage of Mickey's services. Unfortunately, I am the last, since Mickey will be leaving Indiana University in the near future.

To Mickey -- thanks and good luck.

The Director

DIRECTOR'S COMMENTS

In previous years the conference featured papers on a variety of topics, and this year's conference is no exception. A brief glance at the table of contents shows topics which range from the more popular computing applications in accounting, statistics, and administration to the less ordinary application of the computer to the fields of fine arts, medicine, and linguistics. In addition to diversity of subject matter, we were especially pleased to receive a number of presenters from out-of-state colleges and universities. We sincerely hope this trend continues.

Two changes were made from last year's conference. First, the keynote speaker was eliminated to provide more time for the presentations. Secondly, the number of tracks was reduced from 5 to 3 to enable the attendees the opportunity to hear a greater proportion of the presentations of their choice.

We hope that the conference and proceedings will prove informative and beneficial, and that the topics and presentations presented this year will stimulate the interest and imagination to turn many of this year's attendees into next year's presenters.

TABLE OF CONTENTS

Greetings . . . i

Acknowledgments . . . ii

The Director's Comments . . . iii

TRACK 1

Statistical Computing Instruction

✓✓ Wides, Jeffrey W. — Self-Paced Instruction in SPSS With MULTITUTOR . . . 1

Interactive Statistical Computing

✓ Marquette, Jesse F.; DuFala, Margaret M. — ADEPT: The Advanced Data Enquiry Package, Timeshared . . . 12

Data Restructuring

✓ Wakefield, Gordon D. — Facilitating Comparisons Among Common-Response-Category Variables: An SPSS Application . . . 36

Statistical Computing Instruction

✓ Kennedy, Madeleine M.; Lambert, H. David; Stentz, Michael C. — Introducing Statistical Researchers to SPSS: Problems and Prospects . . . 51

Lee, Paul S. T. — A Computerized Demonstration of the Central Limit Theorem in Statistics . . . 59

TRACK 2

Financial Analysis

✓✓ Groomer, S. Michael — The Financial Analysis System: An Integrated Software System for Financial Analysis and Modeling . . . 60

The Computer in Theatre and Journalism

✓ Pepperdine, Warren H. — CAI and the Arts . . . 76

✓ Alexander, George A. — The Computer as Publishing Tool: The IDS System . . . 81

Utility Programs

✓ Dustman, John H. — TIDY . . . 82

Dustman, John H. — EXAMS: Computer Generated Examination System 86

The Computer and Music

Nerheim, Rosalee J. — MIST: A Music Information System. 90
Abeles, Harold F. — Using an EXPER SIM (Experimental Simulation) Model in Teaching Graduate Research Courses in Music Education. 95

TRACK 3

Administration

Barry, Ronald A.; Lane, Malcolm G. — An On-Line Registration System for a Minicomputer Via Event-Driven Fortran. . 96
Jones, Ernest L. — SNAP/SHOT Your Ability to Support that Next Application. 113

Computer Simulations

Roberts, Stephen D.; Schier, James S.; Sulanke, Thom — Modeling and Simulation with INS 138
Shreiner, Scott C. — Simulating American Social Structure: An Interactive Game-Simulation for Career and Life Cycle Decisions. 153

Plotting Applications

Lambert H. David; Childers, Marie E. — Four Dimensional Bar Plotting as an Alternative Data Analytic Methodology. . .169
Alexander, Howard — HYPAT: A Program to Produce Hyperbolic Patterns. 185

Computing Futures

James, Francisco J. — Frontiers in Computer Simulations for Education and Training 196
Goodwin, Gordon; Parsons, Joseph E. — Computer Literacy in the Public Schools. .197

Miller, Paul A. — Project GRAMCORD: A Grammatical Concordance Package Program for the Greek New Testament. 200

SELF-PACED INSTRUCTION IN SPSS WITH MULTITUTOR

Jeffrey W. Wides
Faculty Applications Consultant
Southern Illinois University at Edwardsville

INTRODUCTION

Over the past two or three years the Statistical Package for the Social Sciences has been increasingly utilized at my own University--in undergraduate and graduate courses across several different Schools, for most theses employing statistical analyses and by faculty research. Empirical data and impressionistic conversations indicate that the same thing is happening not only in other universities but also in other computer installations of various types throughout the country.

Yet it has been my repeated experience to realize that most potential statistical users of the computer are both ignorant and scared. Ignorant even of the existence of such statistical packages as SPSS, let alone the basic potentials of its use and the basic mechanics of using it. But fear is perhaps the greater problem, for using SPSS involves a computer, learning something new and using numbers. While those of you who are computer experts may laugh, the fact is that those who have never used a computer often appear paralyzed. This of course creates difficulties for them. On a personal level I would list those difficulties as including a sense of inadequacy, inferiority and failure, an over-dependence on others and wasting one's energy through spinning wheels. On a professional level I would list those difficulties as including falling behind in knowledge, poorer work and inability to complete otherwise reachable goals.

But I do not want to pretend that I am concerned about only THEM. As my University's chief consultant for statistical data processing applications, I neither enjoy nor benefit from dealing with people who are wholly ignorant of statistical packages and paralyzed with fear. Oh to make them less ignorant and less fearful, more independent and more confident! And, I will add, more literate. The manual(s), lectures(s) and workshop(s) are solutions that come to mind, and they are solutions which have been tried--with limited success.

Beginning in February 1978 I began work on another solution-- a series of interactive computer modules intended to teach the novice how to use SPSS. The phenomenon is usually called Computer Assisted Instruction (CAI), and it is intended to enhance the learning process through more effective use of faculty, student and equipment resources. CAI can take many forms, including games, simulations, drill and practice, testing, student inquiry and tutorials. At its best, CAI individualizes instruction according to the student's needs, give the student meaningful control over the learning process, and determines and provides necessary remediation. My lessons are tutorials. They consist of textual materials, review questions, branching routines which are guided sometimes by answers to test-review questions and other items by simple student preference and, finally, actual SPSS runs.

SPI AS AN INSTRUCTIONAL APPROACH

These days we are confronted with a plethora of initials and acronyms, and the computer industry is probably as responsible for the situation as any. To take just this Conference, for example, we are bombarded with, among other letters, CAI, CMI, and CBE. One would think that enough is enough. The idea is appealing, but alas, untrue, for this paper uses still another term. The SPI of "SPI for SPSS" stands for self-paced instruction. We will attempt to explain how the principle of self-paced instruction has been used in the development of a set of interactive teaching modules which are intended to enable the uninitiated to become an independent user of SPSS.[1]

SPI is similar to but in many important respects different from the better known PSI, Keller's Personalized System of Instruction.[2] We can explain the Keller Plan as one which individualizes instruction through allowing the student to control the learning pace, requires unit mastery before proceeding, uses the classroom as a secondary rather than primary teaching medium, stresses written teacher-student communication, and utilizes proctors for grading and tutoring. The Plan includes well-defined objectives for each unit, frequent testing and feedback, and human interaction. Most of the research comparing variates of the Keller Plan with variates of the lecture-discussion method of teaching have shown the former to produce superior academic achievement and retention as well as more positive attitudes toward the learning process.[3]

The main difference between SPI and PSI are the facts that the computer replaces the proctor for tutorial purposes and that the notion of student pacing is more strongly emphasized. The ideas of unitized materials, well-defined objectives, written communication and frequent testing and feedback remain the same. Although unit mastery is not required before a student is allowed to proceed through the SPI for SPSS modules, failure to master earlier material will in fact lead to incomprehension of later modules. This will necessitate retaking the previous materials until they are mastered.

SELF-PACING

As self-pacing may mean different things to different people, I want to clarify how the term is being used. It is being used in the sense that the student has significant control over important aspects of the speed with which s/he is presented with and expected to absorb the learning materials. It does not mean that the learning process is under the total control of the student. That would run the risk of procrastination and no acquisition of skills and understanding.[4] It is expected that such tendencies would be offset by instructors establishing deadlines for assignments and of course the incentive of better grades for assignment completion.

Beginning with a self-evident point, the student controls his/her learning pace through controlling the carriage return. Students can thereby do anything from skimming the materials to carefully reading them several times before asking for more.

The student, interacting with the computer, also controls his/her learning pace through answers to the multiple-choice questions which are asked throughout. This is done in several ways. A wrong answer is sometimes used to pinpoint a problem and clarify the misunderstanding which the answer has denoted. An answer occasionally leads to the student's being taken back to an earlier lesson which has been improperly understood or an earlier assignment which has not been completed. The number of correct answers to a series of questions can determine whether the student is taken to a review unit within a module or allowed to continue his/her linear progress through the module.

A related topic of self-pacing is the student's frequent option to skip non-essential segments of a module. In the first module, for example, s/he may forego more details about logging on, and the student interested only in interactive mode may opt not to take the sections concerned with batch mode. Similarly, a mini-lecture about statistical choice may be ignored.

The learner, furthermore, has choices of reviewing a particular unit or units of a module, thereby controlling the number of times s/he is exposed to the material. Faster learners will not have to review, and slower learners are able to proceed at a private pace consistent with their needs. And as stated above, people who have inadequate mastery of the early modules will be forced by their inability to comprehend later ones to retake early lessons in their entirety.

When a particular module becomes too difficult or boring, the student is given ways to immediately exiting the lesson. On the other hand, when the material is rather easy, the study has opportunity to take two or more lessons in a row before logging off the terminal.

A less obvious but more important way self-paced instruction is implemented in the modules is that the student often controls the order of presentation. The facts that certain segments of the modules can be skipped and that the student has control of what (if anything) is to be reviewed have already been mentioned. Another example would be the module which discusses batch SPSS, where the student is given the choice of initially studying either the necessary program commands or the necessary system commands. Again, the student does not have total control, for the author assures that the basics are covered, and covered incrementally. But where the order of presentation is not essential to the learning process, authority is turned over to the student.

Perhaps the most important way a combination of author-paced and self-paced instruction is reflected by the modules is the principle of incrementally increasing difficulty. Beyond that, the early modules use a stimulus-response approach with maximum possible re-enforcement, whereas the later modules increasingly emphasize student initiative and independence. As the student obtains the actual outputs of frequencies tables, cross-tabulations, product-moment correlations, partial correlation, analysis of variance, regression, and discriminant analysis runs, substantive and theoretical implications are discussed and, most importantly, questions for additional thought and analyses are suggested. The student is steadily given an increasing amount of ability, freedom and encouragement to perform his/her own data analyses. This can be done not only within the modules but also independently of the modules through using the SPSS system file to which the modules are linked. The modules are thus intended to lead the student, at his/her own pace, from a state of relatively passive learning to a state of relatively active inquiry.[5]

The extent to which s/he can carry this inquiry is far-reaching. Having begun with questions suggested but no examined in the modules, the student can use the same data base to examine research questions not even suggested in the text. Less than half of the 26 variables which constitute the illustrative data base are seriously considered in the modules themselves. Similarly, the student is free to go to the pages referenced in the SPSS manual for more details about most topics covered in the modules. It all depends on his/her interests and abilities, and the requirements of the instructor.

Having learned a set of basics about his/her terminal, the system's files and commands, SPSS's data definition, transformation and selection statements, an SPSS sytem file, and, most importantly, performed and interpreted a variety of statistical tasks, the student should come away as a self-sufficient user of SPSS. If the intended results are achieved, and preliminary evidence indicates that this is so, the SPI for SPSS modules will have taught a useful societal skill in an individualized manner. In other words, they will have given the novice some familiarity with computers and some skills in statistical analysis, and they will have done so at a pace with which his/her ability and personality are comfortable.

IMPLEMENTATION

Before specifically illustrating some of those lessons, I will say just a few words about the language in which they are written and the network on which they are stored. The language is MULTITUTOR, a computer based educational approach which was originally developed at Northwestern University and has since been installed at other computer systems throughout the country. The version used for the SPSS lessons includes modifications from Northwestern, Indiana University, the University of Georgia, and the Mid-Illinois Computer Cooperative.

MULTITUTOR utilizes many features of the PLATO IV language, Tutor, but requires only conventional terminals to author and execute CAI lessons. MULTITUTOR is capable of creating and updating CAI lessons, storing student responses, issuing responses contingent on user input, and branching according to a wide variety of user and author-controlled criteria.

MULTITUTOR has the capability both to link the CAI lesson to all or parts of another and to link a CAI lesson to other languages and programs on the system. For example, within MULTITUTOR the user can be led from a MULTITUTOR lesson to a Fortran program and back again to the CAI lesson.

The lessons have been written on a Control Data Corporation Cyber 72-14, which is the central processing unit of an organization called the Mi-Illinois Computer Cooperative. MICC was formed in 1974, and is now comprised of ten state universities throughout Illinois (not including the U. of I.), each of which can lease to other institutions. The schools are connected to the central site through a network of telephone lines and remote terminals, with most of processing being interactive.

CONTENT

As said above, they are primarily directed at the novice. One of their guiding principles is competency-based instruction, i.e. to instruct the student at a level with which his/her competency is comfortable. As you can see from the outline, they begin at an elementary level. The organization is such as to reflect linearly increasing difficulty. Each later module covers a slightly more difficult topic(s) than the preceeding one, and successful completion of each is dependent on successful completion of earlier modules.

This paper will conclude, albeit anti-climatically, with a brief description of the substantive content of the 12 modules. TS1 is simply a general introduction to interactive terminals, the computer, and overcoming one's anxieties about dealing with such machines. TS2 introduces the student to the Mid-Illinois Computer Cooperative, discussing important distinctions among three categories of files and two categories of commands and finally illustrating the use of specific system commands. (TS2A can be taken as a short-cut to reviewing particular sections of TS2.) Still discussing the system per se, TS3 instructs the student increating and storing data files, both interactively and through card reader. The student is also assigned the task for creating his/her own small data file.

The discussion of SPSS begins in TS4, where required data definition statements are presented and the student is taught how to do a small run on his/her previously created file. TS5 introduces some more sophistication and flexibility to required data definition statements, introduces a way of getting statistics from SPSS, and explains how procedural files can be kep for later use.

Although the modules concentrate primarily on SPSS on-line, TS6 is devoted to a presentation of batch SPSS. Here the necessary cards for doing an SPSS run via the card reader are presented, and the minor differences between interactive and batch program commands are illustrated.

The gloves begin to come off in TS7, where the student is presented with a Codebook of our illustrative data file, a DATA LIST statement is detailed, and the handling of missing data is discussed. The data used in modules seven through 12, by the way, are taken from the Center for Political Studies' 1976 National Election Study.

TS8 discusses and illustrates additional optional data definition statements, explains what an SPSS system file is, and takes the student through the process of creating one. In TS9 the student obtains his/her own frequency distributions and cross-tabulations, is given the opportunity to read a mini-lecture on statistical choice, is exposed to a method of reducing response time through sampling the data, and is introduced to data transformation statements. (TS9A can be taken as a short-cut to the mini-lecture.)

TABLE: 1 SPI for SPSS Modules Outline

TS1 INTRODUCTION

- Logging In
- What a Computer Is
- Carriage Return
- User Anxieties
- Machine Problems
- Erasing
- More than One Way

TS2 THE SYSTEM

- Three Categories of Files
- Two Categories of Commands
- Important System Commands
  - Save, Define, Get
  - Attach, Return, Purge

TS3 CREATING A DATA FILE

- Interactively
- Corrections
- Batch
- Student Assignment

TS4 REQUIRED DATA DEFINITION

- Variable List
- N of Cases
- Input Format
- Frequencies Task
- Interactive Line Numbering
- Frequencies Assignment

TS5 MORE TS4

- Names
- To Convention
- Integer Mode
- Statistics
- (Input Medium, Read Input Data, Finish)
- Data List
- More Lines
- Swrite and Sread

TS6 BATCH SPSS

- System Commands
- Program Commands

TS7 OUR DATA

- Codebook
- Data List
- Missing Values

TS8 OUR SPSS SYSTEM FILE

- Var Labels
- Value Labels
- A System File
- Creating It
- Living in a Computer

TS9 FIRST ANALYSES

- Frequencies
- Sample
- Recode
- Crosstabs
- Statistical Mini-Lecture

TS10 GETTING FANCIER

- Pearson Corr
- Compute
- Assign Missing
- If
- Partial Corr
- T-test
- More mean comparisons

TS11 MULTIVARIATE

- Anova
- Select If
- Regression
- Factor

TS12 FINALLY

- More Regression
- Path Analysis
- Discriminant
- Caution and Humility

TS10 introduces and for the most part illustrates, through the student's typing the necessary commands, Pearson correlation, partial correlation, and several ways of comparing means (T-TEST, BREAKDOWN, ANOVA). Additional data transformation statements are also presented, discussed and implemented. (TS10A can be used as a short-cut to a limited review of some of the topics discussed in TS10.)

TS11 illustrates ANOVA and REGRESSION through examining questions of how sex, education and social class relate to beliefs about equal treatment for women. (The discussion of multiple regression can also be accessed via TS11A.) TS12 performs a factor analysis of some "social" and "economic" issue positions, e.g. marijuana and taxes, and it illustrates causal modeling through discussing the effects of income, social class, partisanship and ideology on general warmth or coolness toward Ford and Carter. In addition, there are two discriminant analyses in TS12--one with 1976 vote and the other with region as the dependent variable, and there are brief discussions of sampling, validity, distributional assumptions, and the availability of other analysis techniques. (Selected topics of TS12 can be accessed through TS12A.) TS12 concludes with some words of caution and humility.

I would like to illustrate some of the other principles of the competency-based instruction approach by taking you through the first module. My chief fear is that you will find it too simple, like the economentrician who called the modules condescending. But they are not intended for him. They are intended more for the older faculty member who called them the greatest thing since night baseball. Here I will outline the principles of competency-based instruction which are incorporated into the modules and give some examples of each.

1. Gradually increasing difficulty has already been mentioned--see the outline.
2. Reinforcement
   TS1's "Right on!" "You are an excellent student who will undoubtedly learn how to work the computer." "I understand your concern." "You are of course correct." "You are beautiful." "You are rapidly ending the completion of a successful lesson." The frequency of such reinforcement declines when the student's confidence and sophistication increases as he/she progresses through the modules.
3. Whereas the earlier modules are based on a stimulous-response approach with maximum possible reinforcement, the later modules emphasize a more process-oriented or generative approach to knowledge. There is an attempt to lead the student from a state of relatively passive learning to a state of relatively active inquiry.
4. Humor
   Some of the alternative answers and the responses when those answers are chosen. TS1's "A computer is" "E. a genius." "Anything that cannot think (a computer only follows instructions) cannot be a genius."
5. Wrong answers are frequently used in such a way as to define problem areas for which additional explanation is given. TS2's attempts to clarify types of files and commands and TS4's attempts to clarify the concepts of cases and variables.

6. Simplicity
   TS1's extended analogy between driving a car and running a computer. The avoidance of sophisticated definitional nuances. Continued suggestions of where to go for more in depth discussion, such as particular pages in the manual.
7. Successful completion of assignments are required before moving on. TS3's creating a data file and TS4's FREQUENCIES run.
8. Competency based branching
   TS9 branches to different places depending on how many questions the student correctly answered. There are numerous opportunities for voluntary review within the modules.
9. Illustration
   The specific suggestions for additional analyses made in TS9 through TS12.
10. Learn through doing
   TS9's FREQUENCIES, RECODing and CROSSTABS runs, TS10's variety of SPSS illustrative runs which must be performed by the student. TS11 concentrates on multivariate analysis, including ANOVA, REGRESSION and FACTOR; it also covers the SELECT IF data selection command. TS12 deals in greater depth with regression analysis, illustrates path analysis and finally DISCRIMINANT.

There is no need to go into detail about the precise topics covered by the modules. These are sufficiently covered by the outline in Table 1.

CONCLUSION

I expect the modules to be widely used. During the past summer they were used in a graduate Counselor Education course. They are presently being used in courses taught by the Secondary Education, Psychology, Geography, Economics and Biology Departments. Those courses are at SIUE. Usage growth should continue there, and I expect similar usage at other MICC institutions when the modules will shortly be made available to them. I am also interested in exploring the potentials of the modules being made available to other institutions.

Individuals attending the Conference are welcome to experiment with the TS modules. They can use the terminal or terminals which are already logged in to MICC, and there are sheets beside those terminals explaining how to access the lessons. One word of advice: do not ask for the Codebook to be printed while taking TS7. Copies of the Codebook are already available alongside the terminal(s).

Finally, I would very much appreciate comments or suggestions about improving the modules. If you find any bugs in them, please try to be specific as to which place in which module they were found.

NOTES

1. The modules have been written with financial support from most of the major schools at SIUE, specifically the Graduate School's Faculty Development Program, School of Education, Business School, and School of Science and Technology. Their support also made possible a grant from the U. S. Department of Health, Education and Welfare, and the author is of course grateful for such encouragement.

2. Keller, 1967, 1968; Sherman, 1974.

3. A good summary of such research can be found in Robin, 1976.

4. See, for example, Keller, 1968 and Hess, 1974.

5. This is similar to the generative approach discussed by Kearsley, 1977, p. 13.

REFERENCES

Allison, Elisabeth (1975). "Self-paced Instruction: A Review." The Journal of Economic Education 7:5-12.

Block, James H., ed. (1971). Schools, Society, and Mastery Learning. New York: Holt, Rinehart, and Winston.

Hess, J.H. (1974). "Keller Plan Instruction: Implementation Problems." In J. Gilmour Sherman (ed.), Personalized System of Instruction: 41 Germinal Papers. Menlo Park, Calif: W.A. Benjamin.

Johnston, James M., ed. (1975). Behavior Research and Technology in Higher Education. Springfield, Ill.: Charles C. Thomas.

Kearsley, Greg P. (1974). "Some Conceptual Issues in Computer-Assisted Instruction." Journal of Computer-Based Instruction 4:8-16.

Keller, Fred S. (1967). "Neglected Rewards in the Educational Process." Proceedings of the 23rd Annual Meeting of the American Conference of Academic Deans. Los Angeles.

_____ (1968). "Goodbye, Teacher. . . ." Journal of Applied Behavior Analysis 1:79-89.

_____ and J. Gilmour Sherman (1974). The Keller Plan Handbook: Essays on a Personalized System of Instruction. Menlo Park, Calif.: W.A. Benjamin.

Robin, Arthur L. (1976). "Behavioral Instruction in the College Classroom." Review of Educational Research 46:313-354.

Ruskin, Robert S. (1974). The Personalized System of Instruction: An Educational Alternative. Washington: American Association for Higher Education.

Sayre, John L. and John J. Knightly, eds. (1972). The Personalized System of Instruction in Higher Education: Readings on PSI--the Keller Plan. Enid, Oklahoma: Seminary Press.

Scharf, B. (1975). "PSI--The Only Way to Sail." The Personalized System of Instruction Newsletter 3:2.

Sherman, J. Gilmour, ed. (1974). Personalized System of Instruction: 41 Germinal Papers. Menlo Park, Calif.: W.A. Benjamin.

Wides, Jeffrey W. (1978). "A Proposal to Teach the Novice How to Do It," a paper presented at the 2nd Annual SPSS Users and Coordinators Conference. Chicago, Ill., October 8-11.

Witters, Donald R. and George W. Kent (1972). "Teaching Without Lectures: Evidence in the Case for Individualized Instruction." The Psychological Record 22:169-175.

ADEPT: THE ADVANCED DATA ENQUIRY PACKAGE, TIMESHARED

Jesse F. Marquette
Political Science
The University of Akron

Margaret M. DuFala
Political Science
The University of Akron

I. INTRODUCTION

ADEPT is an integrated conversational statistical analysis package programmed in APL. It provides facile access to a broad selection of frequently used statistical procedures, while its extensive error checking will protect the fumble-fingered user from all but the most inventive mistakes. Data may be entered from a variety of sources, stored in a number of configurations, and program output may be directed to terminal, printer, or stored for further analysis.

II. AVAILABLE PROCEDURES

1. Descriptive Statistics

a. Number of cases, mean, standard deviation, variance
b. Minimum, maximum, and range
c. Skewness
d. Kurtosis
e. Chi square test for normality and probability
f. Number of Z-scores greater than 3 and the largest positive or negative Z-score.
g. Histograms of data grouped:
    1. by specified upper levels
    2. by the even spacing of a specified number of groups over the range
    3. by using multiples of the standard deviation as midpoints of the groups.

2. Correlation

a. Correlation matrix
b. Partial correlations
c. Serial correlations with specified lagging
d. Average correlations across subfiles
e. Canonical correlation

3. Regression

a. Univariate and multivariate multiple regression
b. Stepwise regression with deletion
c. Forward inclusion stepwise
d. Model comparison of the results of multiple regression of subsets against a full model
e. Ridge regression

4. Scattergrams

   a. Plotting of several Y-variables against an X-variable with automatic scaling.

5. Tests on Means

   a. Univariate and multivariate one sample T-tests
   b. Univariate and multivariate two sample (grouped) T-Tests with grouping by:
      1. number of cases in each group
      2. multiple variable classifying
   c. Univariate and multivariate paired variable T-Tests

6. Analysis of Variance

   a. One-way ANOVA and MANOVA
   b. N-way ANOVA and MANOVA

7. Discriminant Analysis

   a. N-group discriminant analysis
   b. N-group classification analysis
      1. test or reduced space
      2. linear or quadratic classification
      3. specified priors

8. Discrete Data Analysis

   a. Contingency tables
      1. Observed frequency
      2. Expected frequency
      3. Columns and/or row marginals
      4. Number of cells with frequency less than 5
      5. Nominal statistics: Chi square and probability, Phi, Cramer's V, Contingency coefficient, and Lambda.
      6. Ordinal statistics: Significance of Kendall's K, Z-value and probability, Tau-B and Tau-C.

   b. Frequency tables
      1. Observed frequency per value, percent of total cases, percent of non-missing cases
      2. Mode, mean, minimum, maximum, range
      3. Frequency histograms

   c. Discrete Multivariate analysis
      1. Frequency table of dependent variable
      2. Multiple prediction to the mode of the dependent variable using multiple predictor variables
      3. Contingency table of observed value to predicted value.
      4. Predictor category correctly predicted among total cases and identification of unit cells.

5. Stepwise prediction to the mode of the dependent variable using all combinations of multiple predictor variables giving lambda, corrected lambda percent correctly predicted and defined number for each step

9. Linear Programming

   a. Algebraic or matrix problem definition
   b. Tableau or annotated solution printing
   c. Ranging (sensitivity analysis)

III. BASIC SYSTEM CHARACTERISTICS

ADEPT is designed to provide a great deal of flexibility with minimum demand on system resources. The package was designed and implemented in a 64K workspace and is adaptable to a 32K workspace.

To achieve flexibility in a limited space, the system uses two complementary approaches to carrying out user assigned tasks.

The first is the familiar concept of an overlay structure. All ADEPT system functions are stored on a system file, and are called, executed, and erased as needed. This minimizes the number of functions actually resident in the active workspace during execution.

The second approach consists of having each user option result in an addition to a program created program. When the user chooses the first option, a basic function FN is initialized, pointing to the correct function in the file to execute the desired procedure. If the user chooses no further option, the task is immediately executed. If several options are chosen, additional code is added to FN, with appropriate pointers and switches.

When the user ceases choosing options, the set of tasks is executed. Since FN is the actual executing program, all option programs and error test routines necessary for input validation and controlling user interaction are also expunged.

This concept leaves, on the average, 55K of a 64K workspace available for statistical procedures, data and calculations. It also makes it possible, and is an ADEPT option, for the user to save a defined task for later or repetitive execution as if he had a deck of cards. Since the system actually requires only 9K for constant resident utility routines, all further increases to workspace capacity, e.g. from 64 to 128K are actually additions to processing area.

IV. PROCEDURES AND VARIABLE SELECTION

From the user's point of view, one of the most important aspects of a statistical package is the ease, (or abstruseness), of task specification. ADEPT is designed to provide the experienced user with rapid access to the desired statistical routines, and at the same time, protect the novice from himself. After signing on and loading ADEPT, the user will be confronted with the message:

```
:OPTIONS?
```

at which point he is free to choose from among the fourteen main options currently available.

The absence of a two-letter code to the left of the colon signifies main option level. If the user was unfamiliar with his choices, a blank response, (carriage return), will produce the list of options available at this level.

Response to the :OPTIONS? query can take three forms:

1. a list of numbers
2. a list of two-letter abbreviations
3. a list of keywords.

Unfortunately, at this time you cannot mix and match responses.

If, for example, the user wanted to activate a data set, perform some correlations and descriptive statistics, he has a number of possible routes. He could request the three options one at a time, getting immediate feedback, or he could respond:

```
2    3    6

SB   C    DS

SET-DATA-BASE    CORRELATION    DESCRIPTIVE-STATISTICS
```

and get the output as an uninterrupted set.

The response lists above will produce, seriatim, the further procedure-specific option lists necessary to identify the data base, the kinds of correlation and regression desired, etc.

A zero, 0, response to any :OPTIONS bumps the user back to the next highest level of specification.

For example, if the user, by some mischance, finds himself confronted with a canonical correlation options query:

```
CC:OPTIONS?
```

A zero response will produce

:OPTIONS

and he can try again.

An inappropriate response to an options query automatically produces a list of available options and a repeat of the query. A second miss produces a repeat of the query. A third produces the list again, etc. At some point we hope the user will either a) get coordinated, b) learn to read, or c) get bored and leave.

The variable selection procedures in ADEPT follow basically the same pattern. When a dataset is initially activated, the user is is asked to select a set of base variables:

e.g. AS:SELECT BASE VARIABLES

to which he may respond with either a list of numbers, a list of variable names, or ALL. Again, the list must be consistent; no mix and match yet.

If the user was working with a 200-variable file, he could select a base set of ten for the particular session. He would then be asked if he wished to see a list of the selected variable names. The list produced consists of the numbers 1 to 10 associated with the corresponding variables. The virtue of this procedure is that should the user choose variables 15 43 129 130....., he can now refer to the set of selected base variables by 1 2 3 4...., or by variable name, whichever he finds most convenient.

Again, a blank response to a :SELECT VARIABLES request will produce the list of active variables, a zero will bump back to :OPTIONS. If the entered list contains misseplling or the variable numbers are out of range, the list will be produced for the first wrong response and the query repeated.

In general, the user's rule for subduing ADEPT is "When in doubt, hit return."

The following example session shows a novice user selecting a data base and generating a correlation matrix and some descriptive statistics.

```
WELCOME TO THE UNIVERSITY OF AKRON A.D.E.P.T. SYSTEM
COPYRIGHT: UNIVERSITY OF AKRON, 1977
```

```
 :OPTIONS?
 0     CANCEL
 1 T  :TRANSFORMATIONS
 2 SB:SET-DATA-BASE
 3 C  :CORRELATION
 4 R  :REGRESSION
 5 S  :SCATTERGRAMS
 6 DS:DESCRIPTIVE-STATISTICS
 7 MT:TESTS-ON-MEANS
 8 DD:DISCRETE-DATA-ANALYSIS
 9 AN:ANALYSIS-OF-VARIANCE
10 DA:DISCRIMINANT-ANALYSIS
11 TM:TASK-MANAGEMENT
12 SS:SUBFILE-SELECT
13 EC:ENVIRONMENT-CONTROL
14 LP:LINEAR-PROGGRAMMING
 :OPTIONS?

  :OPTIONS? 2
SB:SET BASE REQUESTED

SB:OPTIONS?
  0     CANCEL
  1 DB:DEFINE-BASE
  2 RB:RETRIEVE-BASE
  3 AP:APPLY-MAP
  4 SS:SELECT-SUBFILES
  5 SB:SAVE-BASE
  6 SD:STRIP-DATA
  7 OS:ORGANIZE-SUBFILES
  8 AS:ADEPT-FILE-MANAGEMENT
  9 SI:SELECT-IF
 10 SV:SELECT-VARIABLES
 11 CB:CORRELATION-BASE
SB:OPTIONS? 2
SB:RETRIEVE-BASE REQUESTED
SB:ENTER THE NAME OF THE SAVED DATA BASE: SHALD
SB:USING DATA BASE: HALD
SB:LIST VARIABLE NAMES? (Y/N) Y
  1 V1               2 V2               3 V3               4 V4
  5 V5

  :OPTIONS? 3
C :CORRELATION DEFINITION

C :OPTIONS?
 0      CANCEL
 1   CM:CORRELATION-MATRIX
 2   PC:PARTIAL-CORRELATION
 3   SC:SERIAL-CORRELATION
 4   AC:AVERAGE-CORRELATION
 5   CC:CANONICAL-CORRELATION
```

```
C :OPTIONS? 1
C :CM:CORRELATION-MATRIX REQUESTED
C :SELECT VARIABLES
C :
  1 V1              2 V2              3 V3              4 V4
  5 V5
C :SELECT VARIABLES
C : 1 2 3

THE CORRELATION MATRIX IS:

                    V1          V2          V3
V1             1.00000      .22858     ¯.82413
V2              .22858     1.00000     ¯.13924
V3             ¯.82413     ¯.13924     1.00000

  :OPTIONS? 6
DS:DESCRIPTIVE STATISTICS REQUESTED
DS:SELECT VARIABLES
DS: 1 2 3
DS:OPTIONS?
 0      CANCEL
 1   SK:SKEWNESS
 2   KU:KURTOSIS
 3   R :RANGING
 4   CH:CHI-SQ-NORMALITY
 5   EX:EXTREME-Z-SCORES
 6   H :HISTOGRAM
 7   A :ALL
 8   N :NONE
DS:OPTIONS? 1 2



DESCRIPTIVE STATISTICS

                            V1             V2             V3
VALID CASES                 13             13             13
MEAN                   7.46154       48.15385       11.76923
STD. DEVIATION         5.88239       15.56088        6.40513
VARIANCE              34.60256      242.14103       41.02564
SKEWNESS                .60987        ¯.04192         .54155
KURTOSIS               ¯.38012       ¯1.57071       ¯1.36331

  :OPTIONS?
```

An experienced user would have the following interaction, and we list the resulting FN function for reference.

```
WELCOME TO THE UNIVERSITY OF AKRON A.D.E.P.T. SYSTEM
COPYRIGHT: UNIVERSITY OF AKRON, 1977

  :OPTIONS? SB C DS
SB:SET BASE REQUESTED

SB:OPTIONS? RB
SB:RETRIEVE-BASE REQUESTED
SB:ENTER THE NAME OF THE SAVED DATA BASE: SHALD
SB:USING DATA BASE: HALD
SB:LIST VARIABLE NAMES? (Y/N) N
C :CORRELATION DEFINITION

C :OPTIONS? CM
C :CM:CORRELATION-MATRIX REQUESTED
C :SELECT VARIABLES
C : 1 2 3
DS:DESCRIPTIVE STATISTICS REQUESTED
DS:SELECT VARIABLES
DS: SAME

DS:OPTIONS? SK KU
```

With resulting function:

```
      ∇FN[⎕]∇
    ∇ FN;PC;PAR;TR;SW;D̲TD;DTD;TV
[1]   D̲TD←0
[2]   SSV←, 1 2 3
[3]   G̲LG←G̲MG←(ρSSV)ρ0
[4]   →G̲ET 18
[5]   D̲TD←1
[6]   OP← 1 1 0 0 0 0
[7]   →G̲ET 49
    ∇
```

V. DATA STRUCTURE

1. Data Definition

All variables in ADEPT may have an associated variable name, extended description, and a list of missing value codes. If the data is discrete, the variable definition may also contain a set of value labels. The variable names are restricted to 8 alphanumeric characters, the description may be of any length, the missing values are unrestricted, and the value labels may be up to 20 characters.

Variables are defined one variable at a time, e.g.

```
SB:OPTIONS? DB
SB:DEFINE-BASE REQUESTED
SB:FILE DEFINITION
ENTER FILE NAME;FILE DESCRIPTION
EXAMFILE;EXAMPLE FILE DEFINITION SESSION
SB:VARIABLE DEFINITION
FOR EACH VARIABLE ENTER:
VARIABLE NAME;VARIABLE DESCRIPTION:MISSING VALUES:VALUE LABELS
TO CONTINUE ENTRY, END LINE WITH A /
VARIABLE 1:
RESPLACE;RESP. PLACE OF RESIDENCE:98 99:[1]CITY [2]SUBURB [3]RURAL
VARIABLE 2:
SOCLASS ;RESP SUBJECTIVE SOCIAL CLASS ID:98 99 0:[1]LOWER[2]WORKING/
[3]MIDDLE[4]UPPER
```

2. Data Storage

All data are treated as variable as opposed to case vectors. When accessing large files, this considerably reduces I/O time. Data may be stored in one of two forms: an ADEPT Data Base or System File. The Data Base is intended for relatively small data sets and is treated as a single character variable containing the necessary descriptions and data. It is intended for storage and retrieval from a user's workspace.

ADEPT System Files can be treated as either Library files or User Files. A Library file is intended for data sets accessed by numerous users, as during class exercises. Access to a Library file requires only knowledge of the reference number, e.g. 30,48 etc. Access to a user file requires knowledge of the creator's user number and the DSNAME. A user file can be further protected by specifying the file as private, in which case only the creator or a person with supervisor authorization can gain access.

Both the Data Base and System File modes of storage contain all the documentation needed for easy reference and clearly labeled output.

3. Data Entry

Data and associated descriptions may be entered at a terminal, under program control. For large data sets, the system uses a single pass external routine (FORTRAN) to read the descriptions and data may reside on cards, disk, or tape. A subsequent terminal session will initialize the ADEPT System File. At the current moment, the external routine will also accept and flip the data definition information and data from an extant SPSS file. We will shortly provide the facility for the user to go from an ADEPT to an SPSS file.

4. Partitioning Data Files

ADEPT data sets may be grouped into a number of named subfiles at user option. These subfiles may then be accessed individually, in groups, or the subfile structure may be ignored. There is no restriction on the number of cases or variables in subfiles. Each subfile may contain a different number of cases, and/or a different number of variables. When processing multiple subfiles, the ADEPT system automatically verifies the existence of an appropriate match for the selected variable and subfiles. The user may, at his option, impose a temporary subfile structure on a non-subfiled data set, or <u>in lieu</u> of a set of subfiles currently defined.

5. Data Manipulation and Selection

All data modifications in ADEPT are carried on under program control. The permissible modifications available include value recoding, generation of new variables by any permissible APL operator, and generation of new variables using logical conditions for performing transformations. Transformations are defined whenever the user chooses, but executed only when referenced. This makes it possible, and is an ADEPT option, to define and save a set of transformation statements for later or repetitive execution.

Data selection may be accomplished using logical statements, explicit values, or random sampling. These statements are considered transforamtions and are saved with data modification statements at user option.

ADEPT restricts the user to variable numbers as reference in transformation statements. This apparent limitation is actually an added feature of the system. If a user specifies a transformation which sums the values of five variables, he would enter the following code:

```
  :OPTIONS? T
T :TRANFORMATIONS

T :OPTIONS? CT ST
T :CT:CREATE-TRANSFORMATIONS REQUESTED
T :DEFINE TRANSFORMATION 1
T : 10 ← 1 + 2 + 3 + 4 + 5
FOR VARIABLE 10: ENTER
VARIABLE NAME;DESCRIPTION:MISSING VALUES:VALUE LABELS
EXAMTR1 ;EXAMPLE TRANSFORMATION ONE:8 9
T :DEFINE TRANSFORMATION 2
T : 11 ← ¯[5]o6*[0.5]
```

```
FOR VARIABLE 11: ENTER
VARIABLE NAME;DESCRIPTION:MISSING VALUES:VALUE LABELS
EXAMTR2 ;EXAMPLE ARCSIN SQRT OF P TRANSFORM:0
T :DEFINE TRANSFORMATION 3
T :
T :ST:SAVE-TRANSFORMATIONS REQUESTED
T :ENTER NAME TO GIVE TRANSFORMATIONS: EXAMSAVE

  :OPTIONS?
```

The reference numbers used refer to the currently active set of variables. This means that if the defined transformation is saved, at a future terminal session, the user may activate a different data base and execute exactly the same transforamtion. For complex transformations, this is a great time saver.

6. File and Data Base Maintenance

ADEPT permits the creator of a file or data base to edit data definition information, add or delete cases and variables to the file, and to redefine subfile structures. New variables may be added by entry of additional data or by execution of any permissible transformation.

VI. ALGORITHMS AND ACCURACY

The following is a brief discussion of some of the more important technical considerations of ADEPT internal procedures.

1. Correlation Matrix

To minimize non-essential ill-conditioning and insure maximum accuracy for all routines based on the correlation matrix, it is generated as the matrix cross product of the normalized data set. This prior transformation of the data is obviously more time consuming than cumulation of sums of squares and cross products, but we judge the result to be worth the effort.

2. Matrix Determinant

Whenever appropriate, the determinant is generated prior to further operations on a matrix. The test for singularity is set by default to 1E-13, and may be changed by the user. When appropriate, a zero determinant condition is flagged with a list of the variables contained in the matrix at the time of occurence.

3. Matrix Inverse

ADEPT uses the internal matrix inverse operator quad-divide provided with IBM/APLSV. Both the matrix determinant and inverse routines were tested against matrices listed in Gregory, R. T. and Kerney D. L., A Collection of Matrices for Testing Computational Algorithms.

Both routines were considered to be acceptably accurate. Retests will be conducted whenever a new non-IBM implementation is developed.

4. Eigen Structures

ADEPT uses a translation of ACM Algorithm 384-Eigenvalues and Eigenvectors of a Real Symmetric Matrix by G. W. Stewart.

5. Probability Distributions

1-Normal
ACM Algorithm 209 - Gauss by D. Ibbetson
2-Chi Square
ACM Algorithm 299 0 Chi Squared Integral by I. D. Hill and M. C. Pike
3-t
ACM Algorithm 395 - Students t-Distribution by G. W. Hill
4-F
ACM Algorithm 346 - F-test Probabilities by John Morris

This procedure has two options. The actual procedure in ADEPT uses the Morris algorithm for sum of D. F. less than 250, and formula 26.6.15 of NBS Handbook of Mathematical Functions. The numerical approximation is used for sum of D. F. > 250, or when the user specifies a quick and dirty approximation for all procedures.

5-F-inverse

ADEPT currently generates the inverse of F for selected percentage points using algorithm 346 and a brute force approximation method. This procedure is tolerably fast and accurate, but will be improved.

VII. PROGRAM OUTPUT

All output from ADEPT is clearly labeled, including where appropriate, variable names, description, and value labels.

Printed output may be directed to the terminal or the high-speed printer. The route to the printer is installation specific since some installation may not allow direct entry to the print stream.

A number of procedures in ADEPT allow the user to name and save variables containing calculated output for future use.

VIII. EXAMPLES

In the following examples, user responses to system queries have been underlined for emphasis. The general approach in the examples has been to proceed as if the user were familiar with the system to minimize the amount of user interaction displayed.

The first example shows the user accessing an ADEPT library file containing 12 variables and 1490 cases. After established access, the user creates and recodes a variable called SEs1, which is the sum of variables 3, 9, and 10. This is then recoded so that values 2 to 9 become 1, 10 to 14 become 2, and 15 to 20 become 3. The user also recodes variable 7 to delete, (treat as missing data) certain values.

The user indicates that he wants to use all three transformations for the coming task, and specifies that the data are to be selected if variable 11 (race) equals 1 (white).

The user then proceeds to request discrete data analysis, choosing to generate one frequency table and a contingency table.

These interactions and resulting output are listed below:

```
WELCOME TO THE UNIVERSITY OF AKRON A.D.E.P.T. SYSTEM
COPYRIGHT: UNIVERSITY OF AKRON, 1977

  :OPTIONS? SB T SB
SB:SET BASE REQUESTED

SB:OPTIONS? AS
SB:ADEPT-FILE-MANAGEMENT REQUESTED

AS:OPTIONS? LF
AS:LIBRARY-FILE-RETRIEVE REQUESTED
AS:ENTER LIBRARY FILE NO: 30
AS:OPENING DATA FILE
AS:USING DATA BASE: NORC75
AS:SELECT BASE VARIABLE(S)
AS: ALL
AS:LIST VARIABLE NAMES? (Y/N) Y
   1 GOVAID        2 SCHOOLNG      3 DEGREE        4 RELIG
   5 PARTYID       6 VOTE72        7 PRES72        8 POLVIEWS
   9 INCOME       10 CLASS        11 RACE         12 UNEMPLOY
T :TRANFORMATIONS
```

```
T :OPTIONS? CT
T :CT:CREATE-TRANSFORMATIONS REQUESTED
T :DEFINE TRANSFORMATION 1
T : 13←3+9+10
FOR VARIABLE 13: ENTER
VARIABLE NAME;DESCRIPTION:MISSING VALUES:VALUE LABELS
SES1     ;:99
T :DEFINE TRANSFORMATION 2
T : 14←13:(2 TO 9=1)(10 TO 14=2)(15 TO 20=3)
FOR VARIABLE 14: ENTER
VARIABLE NAME;DESCRIPTION:MISSING VALUES:VALUE LABELS
SES2     ;FINAL SES:99:[1]LOW[2]MEDIUM[3]HIGH
T :DEFINE TRANSFORMATION 3
T : 15←7:(8 5=9)
FOR VARIABLE 15: ENTER
VARIABLE NAME;DESCRIPTION:MISSING VALUES:VALUE LABELS
PRES72R ;VOTE FOR MCGOV OR NIXON;0 8 9:L1[MCGOVERN[2]NIXON
T :UNPAIRED [ OR ]
REENTER
PRES72R ;VOTE FOR MCGOV OR NIXON:0 8 9:[1]MCGOVERN[2]NIXON
T :DEFINE TRANSFORMATION 4
T : __
T :SELECT TRANSFORMATIONS BY NUMBER
T : 1 2 3
SB:SET BASE REQUESTED

SB:OPTIONS? SI
SB:SELECT-IF REQUESTED
SI:DEFINE SELECT IF CONDITION
SI: 11=[1]



  :OPTIONS? DD
DD:DISCRETE DATA ANALYSIS REQUESTED

DD:OPTIONS? CT
DD:CONTINGENCY AND/OR FREQUENCY TABLES REQUESTED
DD:SELECT VARIABLES
DD: ALL



DEFINE TABLE REQUEST
DD: 14
DEFINE TABLE REQUEST
DD: 15 14
DEFINE TABLE REQUEST
DD:
DD:OUTPUT SELECTION

DD:OPTIONS? ALL
DD:STATISTICS SELECTION
```

32

```
DD:OPTIONS? 1
DD:FREQUENCIES REQUESTED
DD:PRINT HISTOGRAMS? (Y/N) Y

CONTINGENCY AND/OR FREQUENCY TABLES


TABLE REQUEST     1
------------------

FREQUENCY TABLE FOR: SES2;FINAL SES
----------------

VALUE        SCALE  OBSERVED     PERCENT       PERCENT OF
LABELS       VALUE  FREQUENCY    OF TOTAL   NON-MISSING CASES
----------------

     LOW:      1        345        26.08          27.64
  MEDIUM:      2        646        48.83          51.76
    HIGH:      3        257        19.43          20.59
        :*    99         75         5.67

    VALID CASES:  1248   MISSING CASES:     75

  STATISTICS FOR NON-MISSING CASES
  ----------------

         MINIMUM     MODE     MEAN       MAXIMUM    RANGE
            1          2      1.82          3         2

  FREQUENCY HISTOGRAM
  -------------------

      1       LOW|□□□□□□□□□□□□□□
                 |
      2    MEDIUM|□□□□□□□□□□□□□□□□□□□□□□□□□□
                 |
      3      HIGH|□□□□□□□□□□□
                 |
                  --------------------------

                0                          51.76
                          PERCENT

 CONTINGENCY TABLE FOR: PRES72R;VOTE FOR MCGOV OR NIXON
                       WITH: SES2;FINAL SES

     SES2             PRES72R
 ----------------

  VALUE LABEL|  MCGOVERN:     NIXON:         ROW      ROW
 ACTUAL VALUE|          1          2        MARG.     PCT.
 ----------------

   1:      LOW|OF       59        120        179    22.75
              |EF    58.29     120.71
              |CP↓   23.23      22.81
              |RP→   32.96      67.04
              |TP     7.56      15.38
 ----------------
 2:  MEDIUM|OF       135        259          394    50.51
            |EF    128.30     265.70
            |CP+    53.15      49.24
            |RP>    34.26      65.74
            |TP     17.31      33.21
-----------------
 3:     HIGH|OF        60        147          207    26.54
            |EF     67.41     139.59
            |CP+    23.62      27.95
            |RP>    28.99      71.01
            |TP      7.69      18.85
-----------------
COLUMN MARG.|         254        526
COLUMN PCNT |       32.56      67.44       N = 780
-----------------
THERE ARE   0  CELLS WITH EXP. FREQ. LESS THAN 5
-----------------

NOMINAL LEVEL STATISTICS
-----------------
      CHI SQUARE            DF           SIGNIF
        1.7384               2           .41928
           PHI        CRAMERS V      CONT. COEFF       LAMBDA
         .0472            .0472            .0472        .0000
```

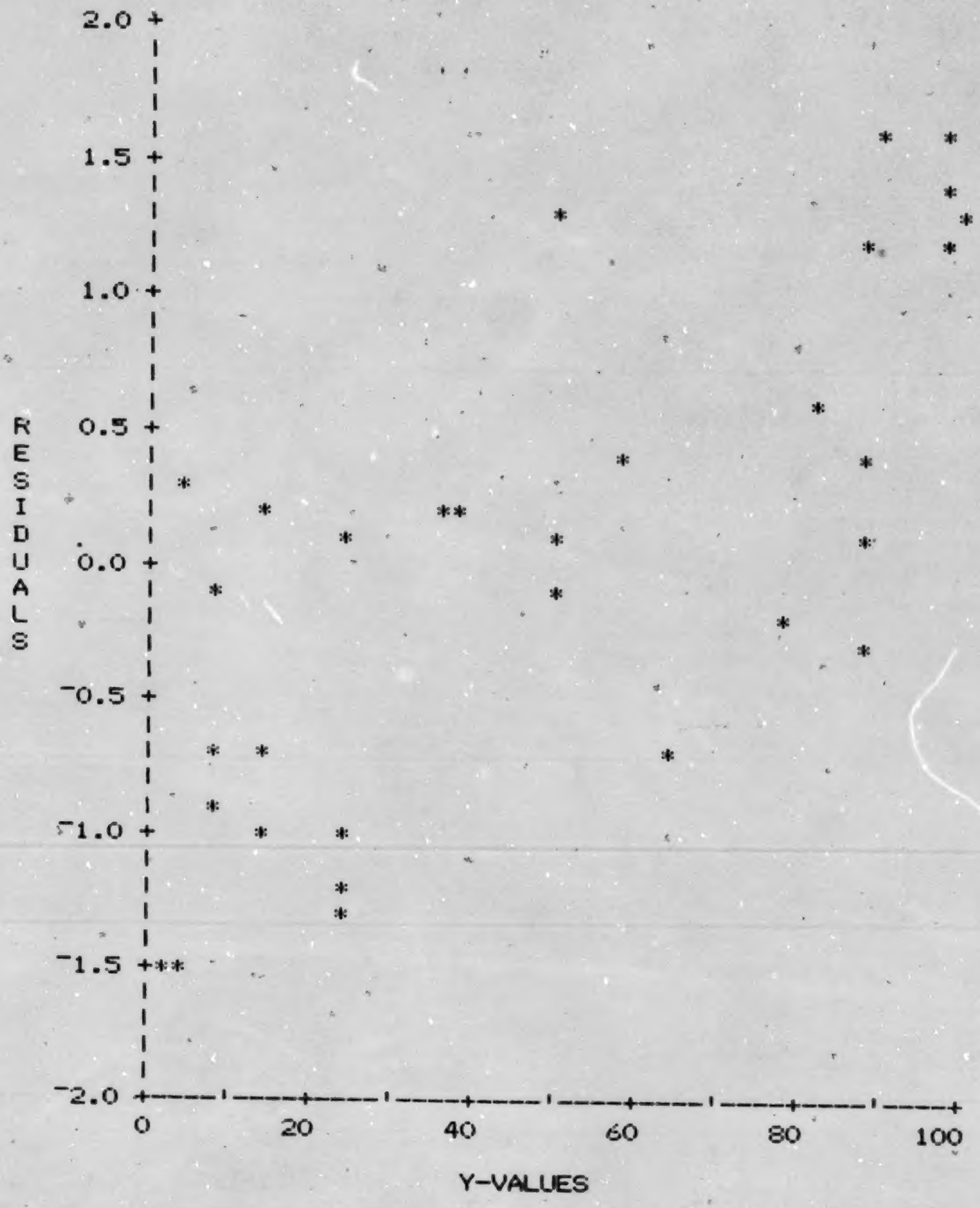
The following example shows a user retrieving a previously saved data base, and a saved task, and simply instructing the system to run the task. The task produces some descriptive statistics on one variable, a correlation matrix and a univariate multiple regression with just a plot of the residuals.

```
WELCOME TO THE UNIVERSITY OF AKRON A.D.E.P.T. SYSTEM
COPYRIGHT: UNIVERSITY OF AKRON, 1977

  :OPTIONS? SB
SB:SET BASE REQUESTED

SB:OPTIONS? RB
SB:RETRIEVE-BASE REQUESTED
SB:ENTER THE NAME OF THE SAVED DATA BASE: TIMMBASE
SB:USING DATA BASE: TIMMDATA
SB:SUBFILE STRUCTURE DEFINED
SB:SELECT SUBFILES BY NUMBER
SB: 1
SB:LIST VARIABLE NAMES? (Y/N) N

  :OPTIONS? TM

TM:OPTIONS? RT
TM:RUN TASK REQUESTED
TM:ENTER NAME OF TASK: TSK1
TM:TSK1 DEFINED ON DATA BASE: TIMMDATA
```

```
          USING SUBFILE 1

            1 HIGHSES;


          DESCRIPTIVE STATISTICS

                                           SAT
          VALID CASES                       32
          MEAN                        47.65625
          STD. DEVIATION              35.32303
          VARIANCE                  1247.71673
          SKEWNESS                      .18791
          KURTOSIS                    ¯1.63337


          THE CORRELATION MATRIX IS:

                                S              NS              NA               SS
          S               1.00000          .30941          .31357           .45617
          NS               .30941         1.00000          .58750           .42885
          NA               .31357          .58750         1.00000           .55645
          SS               .45617          .42885          .55645          1.00000
UNIVARIATE REGRESSION
---------------------

DEPENDENT VARIABLE:          SAT;
INDEPENDENT VARIABLE(S):     S;
                             NS;
                             NA;
                             SS;

MULTIPLE REGRESSION
-------------------

REGRESSION EQUATION: SAT  DEPENDENT
                   'B'        STD. ERR       BETA     STD. ERR
                 COEFFS        OF 'B'       COEFFS    OF BETA     F-VALUES     PROB.
S               2.69201       1.63574       .26683     .16213      2.70846    .10769
NS             ¯3.98746       1.48104     ¯.48406      .17979      7.24867    .01162
NA              5.79280       1.46328       .76857     .19414     15.67186    .00056
SS              ¯.67491       1.24767     ¯.10005      .18495       .29261    .59914
CONSTANT      ¯40.39052

STATISTICS FOR THE ENTIRE EQUATION

ANALYSIS OF VARIANCE
  SOURCE             SS               DF         MS             F          PROB.
REGRESSION     17381.24754             4    4345.31189      5.50867      .00231
ERROR          21297.97121            27     788.81375
TOTAL          38679.21875            31

    MULTIPLE R     MULT. R-SQR     DETER XTRSX
       .67035           .44937           .34267
    AUTOCORR R     DURBIN-STAT
       .13416          1.66537
```

Residuals output may be saved for future analysis, as well as output in tabular and/or plot form as in the following scattergram:

PLOT OF THE NORMALIZED RESIDUALS



The following is an example of a linear programming problem in ADEPT with the first portion showing algebraic problem input, and the second portion showing matrix format input.

The user enters, E to enter, solve, and range a problem. These options could be chosen individually, as is shown in the second portion of the example.

Various tableau printing options are available during solution. The second portion of the example shows the initial and final tableaus being printed.

```
  :OPTIONS? LP
LP:LINEAR-PROGRAMMING REQUESTED

LP:OPTIONS? E
LP:PROBLEM INPUT

LP:OPTIONS? A
LP:ALGEBRAIC INPUT
LP:NAME OF PROJECT?: SAMPLE
LP:MAXIMIZE OR MINIMIZE?: MAX
LP:OBJECTIVE FUNCTION?:
LP: Z=7X+6Y
LP:ENTER CONSTRAINT EQUATIONS, (HIT  RETURN AFTER LAST)
LP:[     1] X+2Y≤400
LP:[     2] 4X+3Y≤1200
LP:[     3] Y≤150
LP:[     4] __
LP:PROBLEM SOLVING
LP:TABLEAU PRINTING

LP:OPTIONS? N

******ANSWERS IN 2 ITERATIONS******

VARIABLE   SOLN. VALUE    MARG. VALUE     BASIC      DUAL
----------------------------------------------------------

VAR1            240.00              0       YES        NO
VAR2             80.00              0       YES        NO
SLK1                 0            .60        NO       YES
SLK2                 0           1.60        NO       YES
SLK3             70.00              0       YES        NO

***MAXIMUM VALUE OF OBJECTIVE FUNCTION IS*** 2160
LP:RANGING PROBLEM
```

```
RANGING LPS

VARIABLES  1 TO 2 ARE THE OBJECTIVE FUNCTION VARIABLES,
AND        3 TO 5 ARE SLACK VARIABLES.
LP:ENTER VARIABLE NUMBERS TO BE RANGED: 1 5 3

***BASIC VARIABLE RANGING FOR  VARIABLE 1***
  VAR     QUOTIENT     FUNCTION         RANGE
   3         ¯1.00         7.00          8.00
   4          4.00         7.00          3.00

***BASIC VARIABLE RANGING FOR  VARIABLE 5***
  VAR     QUOTIENT     FUNCTION         RANGE
   3          ¯.75          .00           .75
   4          8.00          .00         ¯8.00

***CONSTRAINT RANGING FOR VARIABLE 3***
   VAR    QUOTIENT          R.H.S.        RANGE
    2      ¯400.00          400.00       800.00
    1       ¯87.50          400.00       487.50
    5       100.00          400.00       300.00
LP:ENTER VARIABLE NUMBERS TO BE RANGED:
```

```
LP:OPTIONS? I
LP:PROBLEM INPUT

LP:OPTIONS? V
LP:VECTOR INPUT
LP:ENTER M, L, E, G: 2 3 0 0
LP:ENTER COEFFICIENTS FOR LE  CONSTRAINT 1
LP: 1 2
LP:ENTER COEFFICIENTS FOR LE  CONSTRAINT 2
LP: 4 3
LP:ENTER COEFFICIENTS FOR LE  CONSTRAINT 3
LP: 0 1
LP:ENTER 3 RIGHT HAND SIDE CONSTRAINT VALUES
LP: 400 1200 150
LP:ENTER 2 OBJECTIVE FUNCTION COEFFICIENTS
LP: 7 6
LP:ENTER MAX OR MIN: MAX

LP:OPTIONS? S
LP:PROBLEM SOLVING
LP:TABLEAU PRINTING

LP:OPTIONS? IF
```

```
INITIAL TABLEAU

                VAR           VAR          SLK          SLK          SLK
------------------------------------------------------------------------------------
     3         1.00          2.00         1.00          .00          .00      400.00
     4         4.00          3.00          .00         1.00          .00     1200.00
     5          .00          1.00          .00          .00         1.00      150.00
     0        ¯7.00         ¯6.00          .00          .00          .00         .00



TABLEAU AFTER 2 ITERATIONS

                VAR           VAR          SLK          SLK          SLK
------------------------------------------------------------------------------------
     2          .00          1.00          .80         ¯.20          .00       80.00
     1         1.00           .00         ¯.60          .40          .00      240.00
     5          .00           .00         ¯.80          .20         1.00       70.00
     0          .00           .00          .60         1.60          .00     2160.00

******ANSWERS IN 2 ITERATIONS******

VARIABLE   SOLN. VALUE    MARG. VALUE      BASIC      DUAL
----------------------------------------------------------------
VAR1            240.00              0        YES        NO
VAR2             80.00              0        YES        NO
SLK1                 0            .60         NO       YES
SLK2                 0           1.60         NO       YES
SLK3             70.00              0        YES        NO

***MAXIMUM VALUE OF OBJECTIVE FUNCTION IS*** 2160
```

This last example shows a discriminant analysis with the user retrieving a saved data base. The classification matrix printed at the end of a discriminant analysis is always based on all eigenvectors with non-zero eigenvalues, and equal prior probabilities. Other forms of classification analysis can be performed against the saved discriminant output, (in this case FISHDISC). These options can be combined to classify new cases.

```
WELCOME TO THE UNIVERSITY OF AKRON A.D.E.P.T. SYSTEM
COPYRIGHT: UNIVERSITY OF AKRON, 1977

   :OPTIONS? SB
SB:SET BASE REQUESTED

SB:OPTIONS? RB
SB:RETRIEVE-BASE REQUESTED
SB:ENTER THE NAME OF THE SAVED DATA BASE: FISHBASE
SB:USING DATA BASE: FISHER
SB:LIST VARIABLE NAMES? (Y/N) N

   :OPTIONS? DA
DA:DISCRIMINANT ANALYSIS REQUESTED

DA:OPTIONS? 1
DA:SELECT VARIABLES
DA: ALL
DA:SELECT GROUPING VARIABLE(S)
DA: 1
DA:SELECT DEPENDENT VARIABLE(S)
DA: 3 4 5
DA:DO YOU WISH TO SAVE THE DISCRIMINANT OUTPUT? (Y/N) Y
DA:ENTER NAME TO GIVE DISCRIMINANT OUTPUT: FISHDISC
DA:DO YOU WISH TO PRINT THE CASE CLASSIFICATION OUTPUT? (Y/N) N




N-GROUP DISCRIMINANT ANALYSIS

DEPENDENT VARIABLES:  SEPALWD   PETALLT   PETALWD
GROUPING VARIABLE(S):  GROUPID

      TEST FOR EQUAL COVARIANCE MATRICES
         CHI SQ.        D.F.      PROB
       123.46350         24     .00000


       WILK'S TEST FOR EQUAL GROUP MEANS
SOURCE              DET           D.F.    U-VALUE      PROB
GROUPS            .00000            2     .02498     .00000
ERROR         1629.98599          147
TOTAL        65263.26842          149
```

```
GROUP CHARACTERISTICS

VARIABLES             SEPALWD          PETALLT           PETALWD

GROUP        1
--------

MEAN                  3.42800          1.46200            .24600
EFFECT                 .37067         ¯2.29600           ¯.95333
N= 50    P=   .333

V-COV MATRIX

 SEPALWD               .14369           .01170            .00930
 PETALLT               .01170           .03016            .00607
 PETALWD               .00930           .00607            .01111

DET.=         .00004

GROUP        2
--------

MEAN                  2.77000          4.26000           1.32600
EFFECT                ¯.28733           .50200            .12667
N= 50    P=   .333

V-COV MATRIX

 SEPALWD               .09847           .08265            .04120
 PETALLT               .08265           .22082            .07310
 PETALWD               .04120           .07310            .03911

DET.=         .00018

GROUP        3
--------

MEAN                  2.97400          5.55200           2.02600
EFFECT                ¯.08333          1.79400            .82667
N= 50    P=   .333

V-COV MATRIX

 SEPALWD               .10400           .07138            .04763
 PETALLT               .07138           .30459            .04882
 PETALWD               .04763           .04882            .07543

DET.=         .00140

GRAND MEAN            3.05733          3.75800           1.19933
```

```
TOTAL SSCP MATRIX
 SEPALWD          28.30693      ¯49.11881      ¯18.12427
 PETALLT         ¯49.11881      464.32537      193.04574
 PETALWD         ¯18.12427      193.04574       86.56989

BETWEEN GROUPS SSCP MATRIX
 SEPALWD          11.34493      ¯57.23961      ¯22.93266
 PETALLT         ¯57.23961      437.10276      186.77394
 PETALWD         ¯22.93266      186.77394       80.41329

POOLED WITHIN GROUPS SSCP MATRIX
 SEPALWD          16.96200        8.12080        4.80840
 PETALLT           8.12080       27.22260        6.27180
 PETALWD           4.80840        6.27180        6.15660

DISCRIMINANT FUNCTION CHARACTERISTICS

FUNCTION --            1         2
EIGENVALUES     30.14979    .28537
ACCT. VARIAN.     .99062    .00938
CHI.SQ.        538.71926  36.65335
 D.F.            6.00000   2.00000
 PROB.            .00000    .00000

EIGEN
VECTORS
-------

 SEPALWD          .49401   ¯.59178
 PETALLT         ¯.38426    .24833
 PETALWD         ¯.77994   ¯.76689

SCALED
VECTORS
-------

 SEPALWD         2.03457  ¯2.43726
 PETALLT        ¯2.00486   1.29568
 PETALWD        ¯1.93522  ¯1.90285

CLASSIFICATION MATRIX

          PREDICTED GROUP
  ACTUAL            1          2          3
  GROUP
        1          50          0          0
        2           0         48          2
        3           0          1         49

PROBABILITY OF MISCLASSIFICATION
P=        .02000
```

FACILITATING COMPARISONS AMONG COMMON-RESPONSE-CATEGORY VARIABLES: AN SPSS APPLICATION

Gordon D. Wakefield
Academic Computing Coordinator
Indiana University-Purdue University at Fort Wayne

Statement of The Problem

Researchers are frequently interested in comparing the response distributions of several categorical variables each of which has the same response categories. A common example involves comparisons among a group of opinion items each having response categories such as "agree"/"no opinion"/"disagree." Among-item comparisons of the response distributions could be facilitated in this case by a two-way frequency and percentage distribution table containing the response distributions of the items to be compared. Such a table, "response by item," might have as column headings the designations, "Item 1"/"Item 2"/··· /"Item k" (assuming k items), and as row headings the designations, "Agree"/"No Opinion"/"Disagree." Researchers often find it useful to perform comparisons of this sort subject to the control of selected attributes. Three-way (and higher-way) cross-classification tables can be employed to effect the controls required for such comparisons.

A problem facing researchers requiring comparisons of the types described above is that tables which are organized to facilitate such comparisons (via a structure in which the response distributions to be compared reside in a single table) cannot be obtained if conventionally structured data are used. If one's data could be restructured so as to allow the required tables to be obtained from SPSS (or from some other widely available statistical package), the effort involved in among-item response distribution comparisons could be significantly reduced.

Outline of Problem Treatment

This paper describes methods for obtaining tables which facilitate among-variable response distribution comparisons. First considered will be the case where comparisons are not subject to control variables (Case 1). Then the case involving the use of control variables (Case 2) will be explained. These methods involve restructuring one's data so that SPSS can be used to provide the required cross-classification tables. For each of these two cases, the methodology will be presented and an illustrative example provided. Next, suggestions will be offered for simplifying the application of the methods described in this paper. Finally, a summary will be provided.

Methodology for Case 1

Suppose we have a set of categorical variables possessing identical response categories. We shall call such variables "common-response-category variables" or simply, "CRC variables." Our objective is to obtain tables from which comparisons among these variables can be conveniently made.

Through data restructuring it is possible to create for each set of CRC variables a corresponding pair of variables, one of which contains a response code; the other, a number identifying the CRC variable to which the response applies. We shall use the terms"RC (response code or response category) variable" and "VN (variable number) variable" to denote these two variables. A simple example should clarify the relationship between a set of CRC variables and its corresponding variable pair: Given a set of five CRC variables each having the response codes, 1 through 3, the response "3" to the fourth CRC variable would cause the RC variable to take on the value "3", and the VN variable to assume the value "4". The VN and RC variables can be cross-classified to produce the table sought for performing Case 1 comparisons.

Illustration of Case 1

Each of three CRC variables, TVSEARS, TVPENNYS, and TVWARDS (times visited past year: Sears, Penneys, and Wards) has the following response codes and response categories:

Response Codes (Values)	Response Categories (Value Labels)
1	NONE
2	1 TO 12
3	OVER 12

Through data restructuring (to be explained later) the following VN and RC variables were created:

Variable Type	Variable Name	Values	Value Labels
VN	STORE	1	SEARS
		2	PENNEYS
		3	WARDS
RC	TV (Times Visited)	1	NONE
		2	1 TO 12
		3	OVER 12

The variables, STORE and TV, were then cross-classified (using the SPSS procedure, CROSSTABS) to produce Table 1b below.[1]

```
MARKET SURVEY RUN BEFORE RESTRUCTURING -- HYPOTHETICAL DATA

TVSEARS    TIMES VISITED PAST YR -- SEARS

                                   RELATIVE   ADJUSTED      CUM
CATEGORY LABEL          ABSOLUTE     FREQ       FREQ        FREQ
                  CODE     FREQ      (PCT)      (PCT)       (PCT)
NONE               1.        34      11.3       11.3        11.3
1 TO 12            2.       157      52.3       52.3        63.7
OVER 12            3.       109      36.3       36.3       100.0
                          ------    ------     ------
                 TOTAL      300     100.0      100.0

TVPENNYS  TIMES VISITED PAST YR -- PENNEYS

                                   RELATIVE   ADJUSTED      CUM
CATEGORY LABEL          ABSOLUTE     FREQ       FREQ        FREQ
                  CODE     FREQ      (PCT)      (PCT)       (PCT)
NONE               1.        91      30.3       30.3        30.3
1 TO 12            2.       147      49.0       49.0        79.3
OVER 12            3.        62      20.7       20.7       100.0
                          ------    ------     ------
                 TOTAL      300     100.0      100.0

TVWARDS    TIMES VISITED PAST YR -- WARDS

                                   RELATIVE   ADJUSTED      CUM
CATEGORY LABEL          ABSOLUTE     FREQ       FREQ        FREQ
                  CODE     FREQ      (PCT)      (PCT)       (PCT)
NONE               1.       100      33.3       33.3        33.3
1 TO 12            2.       145      48.3       48.3        81.7
OVER 12            3.        55      18.3       18.3       100.0
                          ------    ------     ------
                 TOTAL      300     100.0      100.0
```

TABLE 1a

```
=========================================================================

MARKET SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

   === CROSSTAB OF TV (TIMES VISITED PAST YR) BY STORE ===

                  STORE
           COUNT  I
           COL PCT ISEARS     PENNEYS   WARDS        ROW
                  I                                  TOTAL
                  I        1.I       2.I       3.I
TV                ---------I---------I---------I---------I
              1.  I     34  I     91  I    100  I     225
   NONE           I   11.3  I   30.3  I   33.3  I    25.0
                 -I---------I---------I---------I
              2.  I    157  I    147  I    145  I     449
   1 TO 12        I   52.3  I   49.0  I   48.3  I    49.9
                 -I---------I---------I---------I
              3.  I    109  I     62  I     55  I     226
   OVER 12        I   36.3  I   20.7  I   18.3  I    25.1
                 -I---------I---------I---------I
          COLUMN        300       300       300       900
           TOTAL       33.3      33.3      33.3     100.0
```

TABLE 1b

---

[1] The order of the variables in this cross-classification table ("TV by STORE") was chosen so as to facilitate comparison with Table 1a. The computer printout comprising this and other tables in this paper has been slightly edited so as to achieve greater compactness.

Table 1a, which contains the one-way distributions for the variables, TVSEARS, TVPENNYS, and TVWARDS, has been included for purposes of comparison. This table illustrates the conventional method of handling the three CRC variables: The SPSS procedure, FREQUENCIES, is used in conjunction with conventionally structured data. It will be noted that in this table, the three response distributions are not aligned for convenient visual comparison. Table 1b, on the other hand, displays the response distributions of each "store variable" as a column of the table--thus affording convenient visual comparisons among these distributions.

The reader will note that whereas there were 300 cases before restructuring (Table 1a), there are 900 cases after restructuring (Table 1b). This is because there are three CRC variables. In general, the data restructuring causes the number of cases to be replicated once for each CRC variable.[2] Because of this replication, care must be exercized in interpreting tables obtained from the restructured data. Comparison of Tables 1a and 1b reveals that the frequencies (counts) and percentages in the body of Table 1b are as we are accustomed to seeing them. This is because these frequencies and percentages consist of counts and percentages of respondents. The frequencies and percentages comprising the row and column margins, however, consist of counts and percentages of responses rather than of respondents. Thus, for example, we see from Table 1b that of the 900 responses (300 each to the original variables, TVSEARS, TVPENNYS, and TVWARDS), 449 or 49.9% fell into the category, "1 to 12." Because counts and percentages which are based upon responses are awkward to interpret, it is generally advisable to ignore the row and column frequency totals and their corresponding percentages when interpreting cross-classification tables which have been obtained from the restructured data.

Note that only column percentages were provided in the body of Table 1b. Row percentages and total percentages were suppressed because for this table these percentages would have been based upon numbers of responses rather than upon numbers of respondents. Column percentages were included since they were based upon numbers of respondents. In order to avoid confusion in interpreting cross-classification tables obtained from restructured data, only percentages which are based upon numbers of respondents should be included in the body of the table. The question of what percentages (row, column, or total) to include in the body of a cross-classification table is simplified by the following rule: Include only those percentages which are based upon a single value of the VN variable. For our example, the column percentages were included because each column total corresponds to a single value of the VN variable (STORE). Row and total percentages were excluded since neither the row totals nor the grand total correspond to a single value of STORE.

---

[2] This does not, however, cause the entire original data file to be so replicated. More on this point later.

The data restructuring required for our example will now be explained. If we assume for simplicity that before restructuring, there were only two data cases of the following form:

Variable Name:	TVSEARS	TVPENNYS	TVWARDS
case 1 values:	X_1	Y_1	Z_1
case 2 values:	X_2	Y_2	Z_2

then after restructuring, the data would be as follows:

Variable Type:	VN	RC
Variable Name:	STORE	TV
case 1 values:	1	X_1
case 2 values:	2	Y_1
case 3 values:	3	Z_1
case 4 values:	1	X_2
case 5 values:	2	Y_2
case 6 values:	3	Z_2

Notice that whereas before restructuring there were two cases, after restructuring, there are six (3 times 2) cases. Remembering that the X's, Y's, and Z's represent "times visited" codes for each of the three stores, it is apparent from the form of the restructured data that the cross-classification, "TV by STORE," would produce Table 1b (assuming, of course, 300 rather just two original cases).

The required data restructuring can be performed by a utility program which the user directs by means of the following "restructuring-parameters" card:

```
.R-PARMS        (METHOD=2, CASES=300, VARS=3)
```

Here, "METHOD=2" directs the restructuring utility to employ restructuring method 2: data pooling. The number of cases and variables (before restructuring) are declared to be 300 and 3, respectively. Complete instructions for using the data restructuring utility will be given later.[3] We next turn to Case 2 which involves the use of control variables in among-variable response distribution comparisons.

---

[3] This data restructuring utility which is written in Fortran is available within the Indiana University Computing Network in both batch and interactive forms. Other applications of this utility are described in the following paper: "Enhancing SPSS Flexibility Through Data Restructuring," Gordon D. Wakefield, Proceedings of The Indiana University Computing Network Fifth Annual Conference on Academic Computing Applications, 1978.

Methodology for Case 2

When comparisons of the response distributions of different CRC variables are performed subject to a control variable, these comparisons are made among response distributions which correspond to a particular value of the control variable. In terms of our example, if comparisons among the response distributions of the CRC variables, TVSEARS, TVPENNYS, and TVWARDS are performed subject to the control variable, EDUC (level of education), then these comparisons would be made within a specific level of education. Inspection of Table 2a below reveals that if such comparisons are made subject to level of education 1 (less than high school), it would be necessary to compare the first row of this table ("EDUC by TVSEARS") with the first row of the tables "EDUC by TVPENNYS" and "EDUC by TVWARDS" (These latter two tables are not shown). The fact that the conditional distributions to be compared reside in different cross-classification tables makes for awkward comparisons. If the appropriate conditional distributions could be incorporated into a single cross-classification table, then visual comparisons would be facilitated. Our objective here is to describe how tables having structures which facilitate Case 2 comparisons can be obtained.

```
MARKET SURVEY RUN BEFORE RESTRUCTURING -- HYPOTHETICAL DATA

  *** CROSSTAB OF EDUC BY TVSEARS
  ***   EDUC      LEVEL OF EDUCATION
  ***   TVSEARS   TIMES VISITED PAST YR -- SEARS

                  TVSEARS
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12     ROW
        COL PCT  I                               TOTAL
                 I       1.I      2.I      3.I
EDUC    ---------I---------I---------I---------I
             1.  I    13   I    21   I    10   I    44
  L T HIGH SC    I  29.5   I  47.7   I  22.7   I  14.8
                 I  39.4   I  13.5   I   9.2   I
                 I   4.4   I   7.0   I   3.4   I
                -I---------I---------I---------I
             2.  I    11   I    79   I    48   I   138
  HIGH SC COMPL  I   8.0   I  57.2   I  34.8   I  46.3
                 I  33.3   I  50.6   I  44.0   I
                 I   3.7   I  26.5   I  16.1   I
                -I---------I---------I---------I
             3.  I     5   I    27   I    23   I    55
  SOME COLLEGE   I   9.1   I  49.1   I  41.8   I  18.5
                 I  15.2   I  17.3   I  21.1   I
                 I   1.7   I   9.1   I   7.7   I
                -I---------I---------I---------I
             4.  I     4   I    29   I    28   I    61
  COLLEGE GRAD   I   6.6   I  47.5   I  45.9   I  20.5
                 I  12.1   I  18.6   I  25.7   I
                 I   1.3   I   9.7   I   9.4   I
                -I---------I---------I---------I
         COLUMN       33       156       109       298
         TOTAL      11.1      52.3      36.6     100.0

NUMBER OF MISSING OBSERVATIONS =        2
```

TABLE 2a

If the data file is appropriately structured, and the VN and RC variables are appropriately defined, then tables which facilitate among-variable response distribution comparisons subject to a control variable can be obtained simply by using SPSS to produce appropriate three-way cross-classifications incorporating the VN, RC, and control variables.

Illustration of Case 2

If we extend the example of Case 1 to include the control variable, EDUC, and if the VN variable (STORE) and RC variable (TV) are defined as before, then with appropriate data restructuring, tables which facilitate Case 2 comparisons will be generated by the three-way cross-classification, "STORE by TV by EDUC." For this three-way cross-classification, four two-way tables are produced -- one for each level of EDUC. The first of these two-way tables, Table 2b, is shown below:

```
MARKET SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

  === CROSSTAB OF STORE BY TV -- CONTROLLING FOR EDUC
  ===     STORE
  ===     TV        TIMES VISITED PAST YR
  ===     EDUC      LEVEL OF EDUCATION -- VAL 1   L T HIGH SC

                    TV
            COUNT  I
          ROW PCT  INONE      1 TO 12  OVER 12     ROW
                   I                               TOTAL
                   I        1.I       2.I       3.I
STORE     ---------I---------I---------I---------I
               1.  I    13   I    21   I    10   I    44
  SEARS            I  29.5   I  47.7   I  22.7   I  33.3
                  -I---------I---------I---------I
               2.  I    18   I    18   I     8   I    44
  PENNEYS          I  40.9   I  40.9   I  18.2   I  33.3
                  -I---------I---------I---------I
               3.  I    19   I    18   I     7   I    44
  WARDS            I  43.2   I  40.9   I  15.9   I  33.3
                  -I---------I---------I---------I
           COLUMN       50        57        25        132
            TOTAL     37.9      43.2      18.9      100.0
```

TABLE 2b

This table permits convenient among-STORE comparisons of the TV response distributions within level of education 1 (less than high school). Similar comparisons for the other levels of education may be obtained from the remaining two-way tables (not shown) within the cross-classification, "STORE by TV by EDUC." Comparison of Table 2b with Table 2a reveals the advantage of the "2b" arrangement: The three conditional TV response distributions (for educational level 1) which are to be compared, reside in a single table (Table 2b) rather than in three separate tables of the type depicted in Table 2a.

Because Table 2b was obtained from the restructured data, the same rule of interpretation applies as was given for Case 1: Only the percentages which are based on a single value of the VN variable (STORE) are subject to conventional interpretation. Thus only the row percentages were specified for Table 2b.

Thus far we have considered the Case 2 comparisons consisting of among-STORE contrasts of the TV response distributions controlling for EDUC. We shall next consider Case 2 comparisons consisting of among-STORE contrasts of the EDUC response distributions controlling for TV. These latter contrasts are easily obtained from the two-way tables within the three-way cross-classification, "EDUC by STORE by TV." The first of these two-way tables is Table 2c below:

```
MARKET SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

  === CROSSTAB OF EDUC BY STORE -- CONTROLLING FOR TV
  ===     EDUC    LEVEL OF EDUCATION
  ===     STORE
  ===     TV      TIMES VISITED PAST YR -- VAL 1   NONE

                     STORE
            COUNT  I
          COL PCT  ISEARS     PENNEYS  WARDS        ROW
                   I                                TOTAL
                   I        1.I       2.I       3.I
EDUC      ---------I---------I---------I---------I
                1. I    13   I    18   I    19   I    50
  L T HIGH SC      I  39.4   I  19.8   I  19.2   I  22.4
                  -I---------I---------I---------I
                2. I    11   I    31   I    43   I    85
  HIGH SC COMPL    I  33.3   I  34.1   I  43.4   I  38.1
                  -I---------I---------I---------I
                3. I     5   I    17   I    16   I    38
  SOME COLLEGE     I  15.2   I  18.7   I  16.2   I  17.0
                  -I---------I---------I---------I
                4. I     4   I    25   I    21   I    50
  COLLEGE GRAD     I  12.1   I  27.5   I  21.2   I  22.4
                  -I---------I---------I---------I
           COLUMN       33        91        99        223
            TOTAL     14.8      40.8      44.4      100.0
```

TABLE 2c

Because this table is based upon restructured data, only the percentages which are based upon a single value of STORE--the column totals in this case--are specified. Comparison of Tables 2a and 2c reveals the greater convenience of 2c for making among-STORE comparisons.

The data restructuring required for obtaining Tables 2b and 2c will now be explained. If we assume for simplicity that before restructuring there were only two data cases of the following form:

Variable Name:	TVSEARS	TVPENNYS	TVWARDS	EDUC
case 1 values:	X_1	Y_1	Z_1	E_1
case 2 values:	X_2	Y_2	Z_2	E_2

then after restructuring, the data would be as follows:

Variable Type:	VN	RC	Control
Variable Name:	STORE	TV	EDUC
case 1 values:	1	X_1	E_1
case 2 values:	2	Y_1	E_1
case 3 values:	3	Z_1	E_1
case 4 values:	1	X_2	E_2
case 5 values:	2	Y_2	E_2
case 6 values:	3	Z_2	E_2

Notice that as before, each original case has been converted into three new cases--one for each CRC variable. Notice also that each value of the "control variable," EDUC, has been copied onto all three new cases arising from a single original case. We shall use the terms "carry-along" or "tag-along" as a concise way of expressing this copying phenomenon. The reason for choosing these terms will next be explained with reference to our example.

Notice that both before and after restructuring, the value E_k is associated with X_k, Y_k, and Z_k. This was accomplished by "carrying along" E_k onto all cases arising from original case k. Alternatively, we may say that the association of E_k with X_k, Y_k, and Z_k has been preserved in the restructuring by having E_k "tag along" with X_k, Y_k, and Z_k. The control variable, EDUC, may thus be referred to as a "carry-along" or "tag-along" item in terms of the data restructuring to be performed.

The reader will note that the data restructuring for Case 2 is identical to that for Case 1 except for the inclusion of the tag-along item, EDUC. Thus the Case 2 restructuring can also be used to obtain Table 1b which was used for Case 1 comparisons.

The Case 2 restructuring can be performed by the data restructuring utility program mentioned earlier. For our example, the required restructuring-parameters card is as follows:

```
    .R-PARMS        (METHOD=3, CASES=300, VARS=3, TAGS=1)
```

or

```
    .R-PARMS        (M=3, C=300, V=3, T=1)
```

Here, "METHOD=3" directs the restructuring utility to employ restructuring method 3: data pooling with carry-along. The number of cases and variables (before restructuring) are declared to be 300 and 3, respectively. The designation, "TAGS=1," indicates that there is 1 tag-along item (EDUC). Notice that for purposes of restructuring, EDUC is not termed a "variable" and so is not included in the count, "VARS=3."

Other control cards required by the restructuring utility are as follows: the .PRINTOUT card, the .OLD FORMAT card, the .NEW FORMAT card, and the .FINISH card. The complete set of data restructuring control cards required for our example are shown below:

```
column 1       column 16
.PRINTOUT      LONG
.R-PARMS       (M=3, C=300, V=3, T=1)
.OLD FORMAT    (17X,F1.0,2X,F1.0,2X,F1.0/64X,A1)
.NEW FORMAT    (I3,3X,2I3,5X,F2.0,5X,A1)
.FINISH
```

The key word options for the .PRINTOUT card are "LONG" and "SHORT." The .OLD FORMAT card contains the format of an original case. The .NEW FORMAT card contains the format of a new case. The restructuring utility requires that F-type format specifications be used for "variables," and that A-type specifications be used for tag-along items.

The restructuring utility not only restructures the data obtained from the original cases, it also causes three special items to be generated and automatically included on the new cases. These three items--which are written at the beginning of each new case--are as follows:

(1) The new case number.

(2) The old case number.
This is the number of the original case from which the particular new case was derived.

(3) The position-within-subgroup number.
On the original cases, the three CRC variables form a subgroup. The position of each such variable within the subgroup is the position-within-subgroup number of the variable. In terms of our example, this index is read from the restructured data as the VN variable, "STORE."

The restructuring utility requires that I-type format specifications be used to provide for the writing of these three special items onto to the new cases. Provision for the writing of these three items is required even if they are not subsequently used by SPSS. For our example, only the last of these three items is used by SPSS. From the above .NEW FORMAT card, we see that these three items were written according to the format specification, "I3, 3X, 2I3." A partial listing of the data file after restructuring follows:

```
LISTING OF DATA AFTER RESTRUCTURING -- FILE 'NEW'
--------------------------------------------------

 Not used by SPSS
 in our example.
 New case  Old case
 number    number    STORE    TV    EDUC
    1         1        1      2.     4
    2         1        2      3.     4
    3         1        3      1.     4
    4         2        1      3.     1
    5         2        2      2.     1
    6         2        3      3.     1
                    .
                    .
                    .
  898       300        1      2.     2
  899       300        2      1.     2
  900       300        3      1.     2
```

Simplifying Suggestions

Obtaining all of the tables shown above involved a fair amount of effort: Two computer runs were required--one for the original data (Tables 1a and 2a), and another for the restructured data (Tables 1b, 2b, and 2c). In addition, care had to be exercized in deciding whether row or column percentages should be specified. Listings of the computer jobs for obtaining these tables follow:

LISTING OF JOB 1: SPSS Run Involving No Data Restructuring

```
ABC.
ACCOUNT,9998,PASS.
GET,OLDAT10.
GLIB(SPSS,D=OLDAT10)
7/8/9
RUN NAME         MKT SVY RUN BEFORE RESTRUCTURING--HYPOTHETICAL DATA
VARIABLE LIST    TVSEARS,TVPENNYS,TVWARDS, EDUC
INPUT FORMAT     (17X,F1.0,2X,F1.0,2X,F1.0/64X,F1.0)
N OF CASES       300
VAR LABELS       TVSEARS, TIMES VISITED PAST YR -- SEARS/
                 TVPENNYS, TIMES VISITED PAST YR -- PENNEYS/
                 TVWARDS, TIMES VISITED PAST YR -- WARDS/
                 EDUC, LEVEL OF EDUCATION
VALUE LABELS     TVSEARS TO TVWARDS  (1)NONE (2)1 TO 12
                  (3)OVER 12/
                 EDUC (1)L T HIGH SC (2)HIGH SC COMPL
                  (3) SOME COLLEGE (4) COLLEGE GRAD
MISSING VALUES ALL (0)
FREQUENCIES      GENERAL=TVSEARS,TVPENNYS,TVWARDS
READ INPUT DATA
CROSSTABS        EDUC BY TVSEARS TO TVWARDS
FINISH
6/7/8/9
```

LISTING OF JOB 2: SPSS Run Involving Data Restructuring

```
ABC,T20.
ACCOUNT,9998,PASS.
GET,OLDAT10.
LCALL,DRP3(D=OLDAT10)/UN=9999.
GLIB(SPSS,D=NEW)
7/8/9
.PRINTOUT       LONG
.R-PARMS        (M=3, C=300, V=3, T=1)
.OLD FORMAT     (17X,F1.0,2X,F1.0,2X,F1.0/64X,A1)
.NEW FORMAT     (I3,3X,2I3,5X,F2.0,5X,A1)
.FINISH
7/8/9
RUN NAME        MKT SVY RUN AFTER RESTRUCTURING--HYPOTHETICAL DATA
VARIABLE LIST   STORE,TV,EDUC
INPUT FORMAT    (9X,F3.0,5X,F2.0,5X,F1.0)
N OF CASES      900
VAR LABELS      TV, TIMES VISITED PAST YR/
                EDUC, LEVEL OF EDUCATION
VALUE LABELS    STORE (1)SEARS (2)PENNEYS (3)WARDS/
                TV (1)NONE (2)1 TO 12 (3)OVER 12/
                EDUC (1)L T HIGH SC (2)HIGH SC COMPL
                 (3) SOME COLLEGE (4) COLLEGE GRAD
MISSING VALUES ALL (0)
CROSSTABS       TV BY STORE            |
OPTIONS         3,5                    |
READ INPUT DATA                        |  SPSS Procedure Section
CROSSTABS       STORE BY TV BY EDUC    |______________________
OPTIONS         4,5                    |
CROSSTABS       EDUC BY STORE BY TV    |
OPTIONS         3,5                    |
FINISH
6/7/8/9
```

Job 1 is a straightforward SPSS job which processes the original data. These data reside on the file, OLDAT10. Job 2 invokes the data restructuring utility, DRP3, and passes the restructured data file, NEW, to SPSS for processing.

It is possible to obtain from a single computer job essentially the same information as was produced by the two computer jobs listed above. It is also possible to specify the tables in such a manner as to simplify the decision of whether row or column percentages should be included. The single job which would replace the two shown above would be the same as Job 2 except that the SPSS procedure section would be as follows:

```
CROSSTABS       EDUC BY TV BY STORE
READ INPUT DATA
CROSSTABS       STORE BY TV/
                STORE BY TV BY EDUC/
                STORE BY EDUC BY TV
OPTIONS         4,5
```

The first cross-classification specified in this new procedure section, "EDUC by TV by STORE," will produce three two-way tables (one for each store), the first of which is as follows:

```
MARKET SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

  === CROSSTAB OF EDUC BY TV -- CONTROLLING FOR STORE
  ===    EDUC    LEVEL OF EDUCATION
  ===    TV      TIMES VISITED PAST YR
  ===    STORE
                                          VAL 1    SEARS

                 TV
         COUNT   I
         ROW PCT INONE      1 TO 12  OVER 12     ROW
         COL PCT I                               TOTAL
                 I      1.I      2.I      3.I
EDUC     --------I--------I--------I--------I
              1. I    13  I    21  I    10  I     44
  L T HIGH SC    I  29.5  I  47.7  I  22.7  I   14.8
                 I  39.4  I  13.5  I   9.2  I
                 I   4.4  I   7.0  I   3.4  I
                -I--------I--------I--------I
              2. I    11  I    79  I    48  I    138
  HIGH SC COMPL  I   8.0  I  57.2  I  34.8  I   46.3
                 I  33.3  I  50.6  I  44.0  I
                 I   3.7  I  26.5  I  16.1  I
                -I--------I--------I--------I
              3. I     5  I    27  I    23  I     55
  SOME COLLEGE   I   9.1  I  49.1  I  41.8  I   18.5
                 I  15.2  I  17.3  I  21.1  I
                 I   1.7  I   9.1  I   7.7  I
                -I--------I--------I--------I
              4. I     4  I    29  I    28  I     61
  COLLEGE GRAD   I   6.6  I  47.5  I  45.9  I   20.5
                 I  12.1  I  18.6  I  25.7  I
                 I   1.3  I   9.7  I   9.4  I
                -I--------I--------I--------I
         COLUMN       33      156      109       298
          TOTAL     11.1     52.3     36.6     100.0
```

TABLE 3

Notice that this table contains the same information as was provided by the cross-classification "EDUC by TVSEARS" in Job 1 (See Table 2a). Thus this three-way cross-classification ("EDUC by TV by STORE") takes the place of the three two-way tables specified in Job 1.

The second cross-classification in this new procedure section, "STORE by TV," contains the same information as does Table 1b ("TV by STORE"). The VN variable, STORE, has been placed first, however, as it has also been in the remaining cross-classifications, so as to make the appropriate percentages row percentages for all of these tables.

The third and fourth cross-classifications, "STORE by TV by EDUC" and "STORE by EDUC by TV," contain the same information, respectively, as do Tables 2b and 2c.

The tables obtained from the computer job under discussion have been reproduced on the following page.

Table Ia: From EDUC by TV by STORE

```
MKT SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

  *** CROSSTAB OF EDUC BY TV -- CONTROLLING FOR STORE
  ***     EDUC    LEVEL OF EDUCATION
  ***     TV      TIMES VISITED PAST YR
  ***     STORE                               VAL 1   SEARS

                  TV
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12    ROW
        COL PCT  I                              TOTAL
                 I       1.I      2.I      3.I
EDUC      --------I--------I--------I--------I
              1.  I    13  I    21  I    10  I    44
  L T HIGH SC     I  29.5  I  47.7  I  22.7  I  14.8
                  I  39.4  I  13.5  I   9.2  I
                  I   4.4  I   7.0  I   3.4  I
                 -I--------I--------I--------I
              2.  I    11  I    79  I    48  I   138
  HIGH SC COMPL   I   8.0  I  57.2  I  34.8  I  46.3
                  I  33.3  I  50.6  I  44.0  I
                  I   3.7  I  26.5  I  16.1  I
                 -I--------I--------I--------I
              3.  I     5  I    27  I    23  I    55
  SOME COLLEGE    I   9.1  I  49.1  I  41.8  I  18.5
                  I  15.2  I  17.3  I  21.1  I
                  I   1.7  I   9.1  I   7.7  I
                 -I--------I--------I--------I
              4.  I     4  I    29  I    28  I    61
  COLLEGE GRAD    I   6.6  I  47.5  I  45.9  I  20.5
                  I  12.1  I  18.6  I  25.7  I
                  I   1.3  I   9.7  I   9.4  I
                 -I--------I--------I--------I
          COLUMN      33      156      109      298
           TOTAL    11.1     52.3     36.6    100.0
```

Table II: STORE by TV

```
MKT SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

*** CROSSTAB OF TV (TIMES VISITED PAST YR) BY STORE ***

                  TV
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12    ROW
                 I                              TOTAL
                 I       1.I      2.I      3.I
STORE     --------I--------I--------I--------I
              1.  I    34  I   157  I   109  I   300
  SEARS           I  11.3  I  52.3  I  36.3  I  33.3
                 -I--------I--------I--------I
              2.  I    91  I   147  I    62  I   300
  PENNEYS         I  30.3  I  49.0  I  20.7  I  33.3
                 -I--------I--------I--------I
              3.  I   100  I   145  I    55  I   300
  WARDS           I  33.3  I  48.3  I  18.3  I  33.3
                 -I--------I--------I--------I
          COLUMN     225      449      226      900
           TOTAL    25.0     49.9     25.1    100.0
```

Table IV: STORE by EDUC by TV

```
MKT SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

  *** CROSSTAB OF STORE BY EDUC -- CONTROLLING FOR TV
  ***     STORE
  ***     EDUC    LEVEL OF EDUCATION
  ***     TV      TIMES VISITED PAST YR -- VAL 1   NONE

                  EDUC
          COUNT  I
        ROW PCT  IL T HIGH HIGH SC  SOME COL COLLEGE    ROW
                 I SC      COMPL    LEGE     GRAD       TOTAL
                 I      1.I     2.I      3.I      4.I
STORE     --------I--------I--------I--------I--------I
              1.  I    13  I    11  I     5  I     4  I    33
  SEARS           I  39.4  I  33.3  I  15.2  I  12.1  I  14.8
                 -I--------I--------I--------I--------I
              2.  I    18  I    31  I    17  I    25  I    91
  PENNEYS         I  19.8  I  34.1  I  18.7  I  27.5  I  40.8
                 -I--------I--------I--------I--------I
              3.  I    19  I    43  I    16  I    21  I    99
  WARDS           I  19.2  I  43.4  I  16.2  I  21.2  I  44.4
                 -I--------I--------I--------I--------I
          COLUMN      50       85       38       50      223
           TOTAL    22.4     38.1     17.0     22.4    100.0
```

IVa

```
  *** CROSSTAB OF STORE BY EDUC -- CONTROLLING FOR TV
  ***     STORE
  ***     EDUC    LEVEL OF EDUCATION
  ***     TV      TIMES VISITED PAST YR -- VAL 2   1 TO 12

                  EDUC
          COUNT  I
        ROW PCT  IL T HIGH HIGH SC  SOME COL COLLEGE    ROW
                 I SC      COMPL    LEGE     GRAD       TOTAL
                 I      1.I     2.I      3.I      4.I
STORE     --------I--------I--------I--------I--------I
              1.  I    21  I    79  I    27  I    29  I   156
  SEARS           I  13.5  I  50.6  I  17.3  I  18.6  I  34.9
                 -I--------I--------I--------I--------I
              2.  I    18  I    74  I    27  I    27  I   146
  PENNEYS         I  12.3  I  50.7  I  18.5  I  18.5  I  32.7
                 -I--------I--------I--------I--------I
              3.  I    18  I    70  I    23  I    34  I   145
  WARDS           I  12.4  I  48.3  I  15.9  I  23.4  I  32.4
                 -I--------I--------I--------I--------I
          COLUMN      57      223       77       90      447
           TOTAL    12.8     49.9     17.2     20.1    100.0
```

IVb

```
  *** CROSSTAB OF STORE BY EDUC -- CONTROLLING FOR TV
  ***     STORE
  ***     EDUC    LEVEL OF EDUCATION
  ***     TV      TIMES VISITED PAST YR -- VAL 3   OVER 12

                  EDUC
          COUNT  I
        ROW PCT  IL T HIGH HIGH SC  SOME COL COLLEGE    ROW
                 I SC      COMPL    LEGE     GRAD       TOTAL
                 I      1.I     2.I      3.I      4.I
STORE     --------I--------I--------I--------I--------I
              1.  I    10  I    48  I    23  I    28  I   109
  SEARS           I   9.2  I  44.0  I  21.1  I  25.7  I  48.7
                 -I--------I--------I--------I--------I
              2.  I     8  I    33  I    11  I     9  I    61
  PENNEYS         I  13.1  I  54.1  I  18.0  I  14.8  I  27.2
                 -I--------I--------I--------I--------I
              3.  I     7  I    25  I    16  I     6  I    54
  WARDS           I  13.0  I  46.3  I  29.6  I  11.1  I  24.1
                 -I--------I--------I--------I--------I
          COLUMN      25      106       50       43      224
           TOTAL    11.2     47.3     22.3     19.2    100.0
```

IVc

NUMBER OF MISSING OBSERVATIONS =

Table III: STORE by TV by EDUC

```
MKT SURVEY RUN AFTER RESTRUCTURING -- HYPOTHETICAL DATA

  *** CROSSTAB OF STORE BY TV -- CONTROLLING FOR EDUC
  ***     STORE
  ***     TV      TIMES VISITED PAST YR
  ***     EDUC    LEVEL OF EDUCATION -- VAL 1   L T HIGH SC

                  TV
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12    ROW
                 I                              TOTAL
                 I       1.I      2.I      3.I
STORE     --------I--------I--------I--------I
              1.  I    13  I    21  I    10  I    44
  SEARS           I  29.5  I  47.7  I  22.7  I  33.3
                 -I--------I--------I--------I
              2.  I    18  I    18  I     8  I    44
  PENNEYS         I  40.9  I  40.9  I  18.2  I  33.3
                 -I--------I--------I--------I
              3.  I    19  I    18  I     7  I    44
  WARDS           I  43.2  I  40.9  I  15.9  I  33.3
                 -I--------I--------I--------I
          COLUMN      50       57       25      132
           TOTAL    37.9     43.2     18.9    100.0
```

IIIa

```
  *** CROSSTAB OF STORE BY TV -- CONTROLLING FOR EDUC
  ***     STORE
  ***     TV      TIMES VISITED PAST YR
  ***     EDUC    LEVEL OF EDUCATION -- VAL 2   HIGH SC COMPL

                  TV
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12    ROW
                 I                              TOTAL
                 I       1.I      2.I      3.I
STORE     --------I--------I--------I--------I
              1.  I    11  I    79  I    48  I   138
  SEARS           I   8.0  I  57.2  I  34.8  I  33.3
                 -I--------I--------I--------I
              2.  I    31  I    74  I    33  I   138
  PENNEYS         I  22.5  I  53.6  I  23.9  I  33.3
                 -I--------I--------I--------I
              3.  I    43  I    70  I    25  I   138
  WARDS           I  31.2  I  50.7  I  18.1  I  33.3
                 -I--------I--------I--------I
          COLUMN      85      223      106      414
           TOTAL    20.5     53.9     25.6    100.0
```

IIIb

```
  *** CROSSTAB OF STORE BY TV -- CONTROLLING FOR EDUC
  ***     STORE
  ***     TV      TIMES VISITED PAST YR
  ***     EDUC    LEVEL OF EDUCATION -- VAL 3   SOME COLLEGE

                  TV
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12    ROW
                 I                              TOTAL
                 I       1.I      2.I      3.I
STORE     --------I--------I--------I--------I
              1.  I     5  I    27  I    23  I    55
  SEARS           I   9.1  I  49.1  I  41.8  I  33.3
                 -I--------I--------I--------I
              2.  I    17  I    27  I    11  I    55
  PENNEYS         I  30.9  I  49.1  I  20.0  I  33.3
                 -I--------I--------I--------I
              3.  I    16  I    23  I    16  I    55
  WARDS           I  29.1  I  41.8  I  29.1  I  33.3
                 -I--------I--------I--------I
          COLUMN      38       77       50      165
           TOTAL    23.0     46.7     30.3    100.0
```

IIIc

```
  *** CROSSTAB OF STORE BY TV -- CONTROLLING FOR EDUC
  ***     STORE
  ***     TV      TIMES VISITED PAST YR
  ***     EDUC    LEVEL OF EDUCATION -- VAL 4   COLLEGE GRAD

                  TV
          COUNT  I
        ROW PCT  INONE      1 TO 12  OVER 12    ROW
                 I                              TOTAL
                 I       1.I      2.I      3.I
STORE     --------I--------I--------I--------I
              1.  I     4  I    29  I    28  I    61
  SEARS           I   6.6  I  47.5  I  45.9  I  33.3
                 -I--------I--------I--------I
              2.  I    25  I    27  I     9  I    61
  PENNEYS         I  41.0  I  44.3  I  14.8  I  33.3
                 -I--------I--------I--------I
              3.  I    21  I    34  I     6  I    61
  WARDS           I  34.4  I  55.7  I   9.8  I  33.3
                 -I--------I--------I--------I
          COLUMN      50       90       43      183
           TOTAL    27.3     49.2     23.5    100.0
```

IIId

NUMBER OF MISSING OBSERVATIONS =

These tables are numbered I through IV corresponding to their order in the SPSS job which produced them. All of these tables, it will be recalled, were obtained from the restructured data.

Table Ia (which is the same as Table 3 above, but which is reproduced here for convenience) corresponds to Table 2a, and thus closely approximates the form of the tables that would have to be analyzed in the absence of the techniques described in this paper. Tables Ib and Ic (in which the control variable, STORE, takes on the values "PENNEYS" and "WARDS," respectively) are not shown due to space limitations.

Tables II, III, and IV are of the type which facilitate response distribution comparisons among the CRC variables--for our example, among stores. Table II is appropriate for Case 1 comparisons; Tables III and IV, for Case 2 comparisons. The reader will find it instructive to carefully compare Table Ia with Tables II, III, and IV.

Summary

Comparisons among the response distributions of a set of common-response-category variables are facilitated by the provision of tables which are organized in such a manner that the response distributions to be compared reside in a single table. Tables of this sort can be obtained by using SPSS in conjunction with appropriately structured data. The data structure required to enable SPSS to produce tables of this type is, unfortunately, not the conventional structure. Data restructuring must therefore be performed prior to using SPSS. This restructuring can be performed by the data restructuring utility program described in this paper. In using this utility, a distinction was made between CRC variables and variables which are to be used as controls in certain of the among-CRC-variable comparisons. We used the term "tag-along items" to refer to the "control variables." Although the examples used in this paper involved only a single CRC variable set and a single tag-along item, the data restructuring utility is capable of handling multiple (equal sized) sets of CRC variables as well as multiple tag-along items. For example, the restructuring specification,

```
.R-PARMS     (METHOD=3, CASES=500, SUBGROUPS=8, VARS=6, TAGS=20)
```

could be used to specify the restructuring of a data file from which eight six-member sets of CRC variables and 20 tag-along items are to be read from each of 500 original cases. The number of new cases in this instance would be 3000 (6 times 500).

It should be noted in conclusion that this paper describes but one application of the use of data restructuring in conjunction with SPSS. Other such applications are provided in the paper referenced in footnote 3 of this paper.

Introducing Statistical Researchers
to SPSS: Problems and Prospects

H. David Lambert
Wrubel Computing Center
I. U. Bloomington

Madeleine Kennedy
Wrubel Computing Center
I. U. Bloomington

Michael C. Stentz
Wrubel Computing Center
I. U. Bloomington

Introduction

In the summer of 1978 a group of individuals and departments announced the formation of an informal interest group known as the Indiana University Consortium for Statistical Research Support. Since the inception of our Consortium-related activities the principal organized effort has been to provide instruction in the use of packaged statistical programs on the CDC 6600/172 multi-mainframe system at the Wrubel Computing Center. The bulk of the instruction in this first year has concentrated on the use of SPSS, the Statistical Package for the Social Sciences. We have innovated several aspects of this instruction, and we have some ideas about what future instructional efforts should look like.

Rationale for a New Instructional Format

The growth in demand for activities in this direction are, we think, a valid indicator of the growth of statistical computing at Indiana University. Wrubel Computing Center has traditionally offered free, non-credit instruction in the use of programming languages. Instruction in the use of packaged statistical programs has been a standard offering in the WCC class series. When we first assumed responsibility for WCC's SPSS instruction the classes were held on a twice-a-week basis over a four-week period in one hour sessions. Normally two classes were held during each semester. We were not really satisfied with the opportunity this format afforded us to provide effective instruction. This was so for several reasons:

1. We did not have a prepared set of instructional materials that assured each class would receive a consistent level of instructional treatment.

2. The ad-hoc nature of the course created several problems. Since the class was a "free" class tacked

onto a participant's schedule--which was most likely already overcrowded--student attendance and commitment were erratic. Often Tuesday night's classes would contain a completely different group of students than had attended on the previous Thursday. As the semester progressed, we would very often discover that a starting group of 30 students had dwindled to 6-8 regular participants after four weeks.

3. Although we had only the resources available to offer a single SPSS course, the participants in the class exhibited varying degrees of need. Some students attended for the purpose of extending a previously existing competence while others had never even used a keypunch--although both were equally in need of good instruction.

4. The existing instructional set-up provided us with little opportunity to supervise computing activities of the class participants, even though the particular pedagogy we sought to utilize was very dependent upon students' learning by doing. The experience of designing a good sequence of instruction centered around providing feedback to students about computer runs they had already done. Our discovery that the assigned runs had been done by only one student proved to be a frustration on several occasions.

An initial reaction to these frustrations might be to lay the blame on the lack of motivation on the part of the students. However, lengthy discussions with several participants convinced us that this was really not the case. The problems were inherent in the mechanisms of instruction and not the students.

A New Instructional Format

In January 1978 we revised our SPSS instructional methodology in several ways. These changes greatly enhanced our ability to provide better instruction. The results confirmed our hunch that students would take advantage of our instructional efforts if they were more organized and convenient. The principal change that was instituted was the creation of the "Intensive Weekend" of Introduction to Packaged Statistical programming. These workshops differed from our earlier classes in more than their scheduling. They included a commitment to several elements that are important in providing good instruction.

1. Given our feelings that a consistent instructional treatment was a pre-requisite to good instruction, a workbook was created which included basic instructional materials that we had found helpful in our earlier classes. The workbook laid out a series of tasks and activities which each student was expected to complete.

2. Great efforts were made to provide in-class instruction that was interesting and motivating for the students. These activities included the preparation of audio-visual materials and the use of instructional techniques that promote student involvement in the classes.

3. Since the majority of the students in our intensive weekends are computing novices, substantial efforts were directed at helping them get over their often exaggerated fears of computing. SPSS itself proves a good ally in this pursuit, for it is very easy to create and successfully execute simple jobs. By the end of the 16-20 hours of weekend activity each student has probably run 20 jobs and has experienced success on at least six of them. The great proportion of our activities are directed at helping the slower students in the class. The structured workbook we use makes it possible for the quicker students to proceed at their own pace—or to spend time helping out their more "befuddled" classmates.

As is so often the case in computing instruction, our student participants are not thrown out into the computer center to sink or swim. Their computing activities during the two days are closely supervised. The chief concern of the instructors and assistants is to see that students not be allowed to become completely "derailed" by their frustrations. Instead of being chastised if they make errors in their runs, students are encouraged to share their errors with the group; substantial time is then spent during the formal classroom sessions talking about the sources of SPSS and KRONOS error messages.

4. At about the same time the intensive weekend seminars were instituted with a fixed instructional sequence, the decision was made to offer additional classes at the Intermediate and Advanced levels. Initially the four-week structure was used with these classes, but during the 1978-1979 school year, these too were altered to the intensive, weekend format.

The Intermediate intensive class is basically an extension of the Introductory class. It extends its emphasis onto the interfaces between SPSS and the KRONOS operating system; extends its treatment of SPSS statistical procedures and data management capabilities; and covers special topics like the use of interactive facilities, system level utilities, and basic plotting applications.

The advanced intensive strays somewhat from an emphasis on "programming" per se and extends into a conceptual treatment of statistical topics—specificaly general linear model applications of SPSS procedures T-TEST, ONEWAY, ANOVA, REGRESSION, and MANOVA. Clearly, this new approach to instruction in packaged statistical programming has proved superior to the traditional techniques.

The Instructional Schedule

In the 13 months since we held the first SPSS Intensive Introductory weekend, our best estimate is that approximately 300-350 students and faculty members have participated in these sessions. In most cases the weekend was the first introduction to computing for the participant. Although we have no figures to support this contention, we think that the majority of these students have since put their SPSS skills to use on their own applications. Many of the students and faculty members we consult with now have had the intensive course in the last year—a fact that makes our consulting job much easier. Table 1 provides insight into the types of students who participate in our introductory class.

During spring, 1978 we held three introductory intensive workshops and one six-week introductory class. We offered one six-week intermediate class and an 8-week advanced class. During summer, 1978 we held one introductory intensive workshop. During the Fall semester our instructional schedule included four introductory sessions. Three of these were general sessions open to the entire community. The fourth session was arranged specially for students in the Department of Sociology's graduate statistics course. Attendance for this weekend session was a mandatory element of a graduate-level statistics course in the Sociology Department. The professor participated in this session along with his students. The success of this arrangement causes us to think that this might be a suitable way to structure much of our future instructional activity. In addition the four introductory sessions, two intermediate weekend sessions and one advanced weekend seminar were offered. The weekend intensive format exceeded all of our instruction expectations. The dropout rate is almost zero, the participants are highly motivated, and our instruction is more effective.

Our load for the spring semester of 1979 will remain at the same level as last semester with 3 introductory intensives, 2 intermediate intensives, and 1 advanced intensive. As of February 1, 1979 maximum enrollment has been reached in all sessions except for the second intermediate class. All three scheduled introductory sessions have been filled since the second week of January. While we have exhausted our resources with those weekends already scheduled, we are still receiving 3-5 calls per day from individuals seeking admission into the introductory sessions. We do not anticipate scheduling an additional session until the 6-week summer term.

In addition to these activities we are receiving numerous requests to interface our instruction with that provided in graduate statistics and research classes. During the first two weeks of the semester we were invited to take over the laboratory sessions of a Graduate level Statistics class in the Biology Department. We were able to cover slightly over half of the activities of our standard introductory sessions in the time allocated, and students in the class have been applying what they learned via weekly programming tasks assigned by the professor for the remainder of the semester. In addition, we have scheduled two full Saturday sessions on TELEX use, file manipulation, and SPSS programs MANOVA and G3SLS for the students in the second semester of the graduate statistics class in the Sociology Department. In addition, a half-day session has been requested for a research class in the Graduate Library School. A day-long workshop in using SPSS MANOVA and related routines has been held for the faculty from several departments at IUPU-Fort Wayne.

Future Directions

This seemingly unending demand for instruction in the use of packaged statistical programs provides a unique demand for services that are best provided through Wrubel Computing Center. We have been pleased by the support we have received from Wrubel for implementing our ideas about effective instruction in computer use. However, we are not confident that the current commitments of recources are sufficient to deal with the demands of the next several years. We think it is important to begin now to implement strategies for instruction during the coming year. What follows is a brief exposition of the activities we believe are important future endeavors.

At the top of our agenda is an idea that we have talked about for the last two years. While we are pleased with the format of and reaction to our intensive seminars, they are demanding and exhausting person-intensive activities. We feel that the type of instruction we provide is really best provided in an individualized learning environment. The best way to systematically meet these university-wide instructional needs would be via a permanently staffed learning laboratory characterized by:

1. Computer assisted instruction via MULTITUTOR (or CDC PLATO in the future) lessons which are interfaced with SPSS-ONLINE. This environment would allow students to proceed thorugh a sequence of instruction at their own pace and be exposed to lessons of varying levels of difficulty. While many of these lessons would be developed by local staff there are already "canned" lessons available which could be utilized.

2. Availability of instructional assistance for a large part of the week—including "low-demand" times like evenings and weekends. This assistance would be provided by full-time staff and graduate assistants.

3. A remote RJE station for batch processing of instruction jobs.

4. A large, well equipped classroom for holding organized instructional activites and intensive weekend seminars.

We think there is sufficient need within the University for this type of facility. We also think that there is probably sufficient faculty and student support to at least begin to implement a limited learning laboratory. We have discussed our ideas with several people on campus who are involved in instructional development and in computer assisted instruction. (DDSP). These individuals feel that there are innumerable opportunities to interface the laboratory with other instructional projects on campus. They have also suggested that such a project, if developed properly, could attract sources of outside funding. The realization of the Laboratory is one project that we think is well worth the investment of Wrubel resources.

Another project that we would like to initiate during the coming year is related to the development of useful instructional materials and documentation. While the C.A.I. focus of the laboratory would be a priority there are other needs as well. We would like to continue to devote resources to the refinement of the instructional materials we use in our weekend intensives.

We would like to launch a project designed to reduce our consulting and instructional load by developing, and making available, a series of annotated workbooks aimed at assisting users in actually interpreting and understanding the output of SPSS statistical procedures - particularly those commonly used multivariate procedures, FACTOR, REGRESSSION, DISCRIMINANT, CANONICAL, G3SLS and MANOVA. We think these workbooks would be worth the effort to our users—students and faculty members. Commitment of Wrubel resources to this project could be minimized by developing interfaces with graduate statistics classes on campus, with students providing much of the manpower for the project and their professors serving as evaluators of the final products.

Clearly Wrubel's assumption of an instructional role in the University has been welcomed by both students and faculty members whose departments are unable to fully meet their data analytic instructional needs. We think it is in the best interest of both Wrubel and the University to extend these activities during the coming years. This short paper represents an excerpt from a longer position paper on the future of statistical computing support at Indiana University. Copies of this paper are available from the authors.

WCC PACKAGED PROGRAM CLASS
OUTLINE OF CLASS ACTIVITIES

Seq.	Class Activity	Task for Next Class
No.1	Introduction The typical SPSS job.	Task 1- Punch and run first two example SPSS jobs from intro section of workbook.
No. 2	Discuss runs. Write SPSS program to get frequencies on class data set. Use computer cards which have been punched with the prequiz data for the class.	Task 2- Frequencies on class data set. Read Parts I.,II.A. in Comp WCC Read First 3 sample runs in SPSS runs.
No. 3	Discuss runs. Discuss format specifications. Run the same deck as before but this time specify the social security number to be read with an "A" format specification. Add a "Print Formats" card to tell SPSS how to print the social security number. Add "Var Labels" and "Value Labels" cards for all variables. Declare "Missing Values" for all variables.	Do Task 3. Read SPSS runs 4 and 5.
No. 4	Discuss runs. Write program to convert class data to correct-incorrect form and computer a total score. Don't recode the "No Response" answers to zero as if they were incorrect.	Do Task 4. Read SPSS runs sample run 6. Read COMP WCC Parts II.B., IV.A., IV.B.
No. 5	Discuss runs. Write program to take care of missing data in total score, and to compute total score in various ways. Discuss P-files. Run with data and SPSS input cards on permanent files.	Do Task 5. Read SPSS runs sample runs 7,8. Read COMP WCC Part V.
No. 6	Discuss runs. Application exercise -- given data file, codebook, and 4 questions, write the appropriate program and answer the questions.	Do Task 6.

APPENDIX

TABLE 1

INTRODUCTION TO PACKAGED STATISTICAL PROGRAMMING
1978-1979

TABULATION OF PARTICIPANTS BY DEPARTMENT

DEPARTMENT	PERCENT	DEPARTMENT	PERCENT
BIOLOGY	8.6	ED-IDPYC	1.1
SOCIOLOGY	8.1	ED-READING	1.1
PSYCHOLOGY	7.5	ED-SCIENCE	1.1
BUSINESS	5.9	LIBR SCIENCE	1.1
POLITICAL SCIENCE	5.4	MASS COMMUNICATIONS	1.1
HPER	4.8	NURSING	1.1
SPEA	4.8	TELECOMMUNICATIONS	1.1
EDUCATION	4.3	EAST ASIAN LANGUAGES	.5
ED-PSYCHOLOGY	4.3	ED-ADMINISTRATION	.5
ED-IST	3.8	ED-ADULT	.5
JOURNALISM	3.2	ED-COUNSELING	.5
GEOLOGY	2.7	ED-CSPA	.5
MATH	2.7	ED-INQUIRY	.5
ECONOMICS	2.7	ED-APP LINGUISTICS	.5
MUSIC	2.2	ED-VOCATIONAL	.5
DDSP	1.6	FOLKLORE	.5
ED-HIGHER	1.6	HISTORY	.5
HOME ECONOMICS	1.6	LINGUISTICS	.5
IUPUI	1.6	RECORDS	.5
SPEECH COMMUN	1.6	RESIDENCE LIFE	.5
WCC	1.6	INSTITUTE SEX RSCH	.5
ANTHPOLOGY	1.1	TV AGENCY	.5
COMP SCIENCE	1.1	NONDEGREE	.5
ED-CURRICULUM	1.1		

A COMPUTERIZED DEMONSTRATION OF THE CENTRAL LIMIT THEOREM IN STATISTICS

Paul S.T. Lee, Ph.D.
Senior Research Associate
University of Nebraska at Omaha

The Central Limit Theorem is one of the most important concepts in statistics. It provides a link between sample statistics and population parameters. It is a basic concept for understanding various hypothesis-testing techniques such as Student's t-distribution, X^2-distribution and F-distribution.

Without using computer facilities, it is almost impossible to verify this theorem in the classroom, particularly to non-math majors. As a result, students and instructors alike often become confused and frustrated in their attempt to understand this fundamental part of the study of statistical inference.

For the past two decades, the use of computers in education has stimulated a great deal interest and computer facilities have been made easily accessible to both students and instructors. Computer softwares have been prepared for use by both math and non-math majors.

The paper presents findings from an educational research project dealing with the development of computer softwares capable of conducting sampling experiments for the demonstration of the Central Limit Theorem. Both graphs and tables will be used extensively in presenting the results.

THE FINANCIAL ANALYSIS SYSTEM: AN INTEGRATED SOFTWARE SYSTEM FOR FINANCIAL ANALYSIS AND MODELING

S. Michael Groomer
Assistant Professor of Accounting
Indiana University

During the past fifteen years there has been an emergence of many large machine readable data bases. These data bases have been complemented by the tremendous advancements in computer technology and the utilization of automated procedures for financial analysis and display. Moreover, there is an ever increasing trend toward the utilization of computer-based financial data analysis in the educational curriculum, as well as for financial and industrial concerns.

There are, however, several factors which limit the role of the computer in concert with these data bases as a standard part of the curriculum. While there is a widespread utilization of computer based analyses and modeling in academic research, the same is not true for general classroom instruction and teaching methods. This is due, in part, to inadequate resources such as computer hardware, software and technical support. In addition, students enrolled in courses which treat the fundamentals are seldom able to make use of computer based methods due to their limited or non-existent background in computer programming, computer applications and statistical methods.

A number of generalized statistical packages, such as the Statistical Analysis System (SAS) or the Statistical Package for the Social Sciences (SPSS) have removed some of the barriers associated with student utilization of large data bases. In nearly all cases, however, the desirable display, analysis and modeling capabilities are not directly available. Examples of these deficiencies would include financial statements and comparative analyses. Moreover, there are significant staff requirements necessary to develop, operate and maintain associated data, particularly in the situation where multiple large data bases are utilized. Certainly, there is a need to implement generalized software specifically directed toward the larger and more significant financial data bases.

One such software package recently available for use with the COMPUSTAT[1] financial data is the Financial Analysis System. Thus, the purpose of this paper is to describe the Financial Analysis System (FAS)[2] in facilitating effective utilization of the COMPUSTAT data base in the classroom environment.

67

FAS AND ITS ORIGINS

FAS is a batch-oriented, modular software package of analytical modeling and display routines designed to facilitate the effective utilization of the COMPUSTAT data files. Although relatively new, FAS is being implemented at both academic and commercial installations. The COMPUSTAT data service consists of a number of machine readable libraries of financial, market and statistical information covering several thousand industrial and non-industrial companies. The data is available via lease agreement from Standard and Poor's Compustat Services, Inc. on an annual, quarterly or monthly update basis. Data on the largest and most significant companies listed on the New York, American and Regional Stock Exchanges as well as Canadian companies are available. In addition to broad company coverage, the annual industrial files contain detailed key balance sheet, income statement and market items by individual companies. FAS may be used to access annual data found on the Primary, Supplementary, Tertiary, Over-the-Counter and Canadian Files.

FAS is a software product of the Public Affairs Information Service (School of Business and Public Administration) of the University of Missouri-Columbia and is currently under exclusive license to Management System Developments, Inc. FAS was developed at Missouri after three years of experience and support of several programs developed for use with the COMPUSTAT data which had specific and limited capabilities and required a reasonable amount of computer expertise to use. Thus, FAS arose out of a need to have a broadly based piece of software for display and analysis of COMPUSTAT data which could be more easily utilized, maintained and enhanced.

OPERATIONAL CHARACTERISTICS

FAS is written in a highly portable set of FORTRAN IV consisting of approximately 11,500 lines of code and has been successfully executed on IBM, CDC, UNIVAC and BURROUGHS mainframes. Because of its size, FAS has been overlayed at a majority of the installation sites. Moreover, most implementers have found catalog procedures very helpful in facilitating user access to this software.

Communicating with FAS is easily accomplished via two types of user control cards: procedure cards and request cards. Key word procedures are used to tell FAS which display or analytical routines are to be used. The request cards indicate which companies in the data file are to be selected for processing. Procedure cards and related request cards comprise a request set. One or more request sets may be processed in a given job. The procedures currently available through FAS are summarized on the next page.[3]

COMPUTE: The COMPUTE procedure enables the user to create user defined variables. These can be single variables (scalars) or time-series variables (vectors). Using built-in functions, the user can compute the mean, standard deviation, geometric growth rate, and arithmetic growth rate for time-series variables.

AVG: The AVG procedure computes and displays average industry ratios from a set of 30 predefined ratios.

GROUP: The GROUP procedure instructs FAS to aggregate all of the COMPUSTAT data items for the requested companies. This allows the user to create industry or composite type data.

INPUT: The INPUT procedure enables the user to merge non-COMPUSTAT data with data from the COMPUSTAT files. For example, economic data such as GNP, or additional company data can be read in and used in subsequent computations.

LIST: The LIST procedure instructs FAS to print all or part of the data for the requested companies in a tabular form. Through the use of the ITEMS and YEARS parameters, the user can select specific items or years. If neither ITEMS nor YEARS is specified, all items for the last ten years are printed in an organized data list format--all items grouped by type; i.e., income statement, balance sheet, etc.

OUTPUT: The OUTPUT procedure enables the user to write out, to any peripheral device, user specified data in a user specified format. For example, computed variables such as ratios along with COMPUSTAT data items could be written onto another file to be processed by other user software.

PLOT: The PLOT procedure produces a two-dimensional plot of two user specified variables. The number of rows can be controlled with a ROWS parameter up to a maximum of fifty rows.

PUNCH: The PUNCH procedure enables the user to write to an auxiliary device, an eighty column character record,--normally directed to a card punch. This procedure does not require a user specified format and is provided as a convenience to the non-programming user.

RATIOS: The RATIOS procedure is used to request the default set of 30 predefined financial ratios or a user specified subset. Liquidity ratios, asset ratios, efficiency-ratios, profitability ratios, and leverage ratios comprise the 30 predefined ratios.

RECORD: The RECORD procedure is similar to the PUNCH procedure; however, the data record is written in binary (via an unformatted FORTRAN WRITE).

REGR: The REGR procedure allows the user to estimate multiple regression models using ordinary least squares. The correlation matrix and related descriptive statistics can optionally be printed.

REPORT: The REPORT procedure is used to generate any one of three standard reports to display user specified data. Report layouts provide flexibility for the display of data for multiple years, data items, and companies.

SCREEN: The SCREEN procedure enables the user to select companies based upon user specified criteria. Examples would be selecting only those companies from an industry that had sales in specified years greater than a given amount, selecting companies that had a P/E ratio in the latest year greater than a user specified value.

STATEMENTS: The STATEMENTS procedure is used to generate income statements, balance sheets, and/or source and use of funds statements for specified years for the requested companies.

TITLE: The TITLE procedure permits the user to print an 80 character title at the top of each page of the FAS output.

TREND: The TREND procedure instructs FAS to prepare a trend analysis on a predefined set of variables or a user specified set of variables. The trend is estimated using ordinary least squares. Both a linear trend and compound growth rate model are estimated for the requested variables.

These procedures can be qualified for greater flexibility, through the use of keyword parameters. For example, YEARS is used to specify a specific year or years for time series variables.

In order to make effective use of FAS, the user must be familiar with the annual data record found on the industrial file(s). An individual company record contains the following information.

1. Industry Number
2. CUSIP Company Number (Issuer Code)
3. CUSIP Issue Number and Check Digit
4. File Identification Code
5. Exchange Listing and S&P Index Code
6. Industry Name (Alphabetic Field)
7. Company Name (Alphabetic Field)
8. Stock Ticker Symbol (Alphabetic Field)
9. Fiscal Year End Month of Data for Each of 20 Years
10. Data Year
11. Special Treatment Footnote Code for Each of 20 Years
12. Update Code (0,1,2 or 3)
13. Annual Footnotes. Data Array of 35 Footnotes for 20 Years
14. Data Array of 135 Items for 20 Years

FAS reads the entire record for each company requested. The user can address selected parts of this record through the use of reserved words. For example, the data matrix is accessed with the reserved word D. Thus, a specific item for a specific year would be D(2,75), indicating that the second data item for 1975 is desired for utilization in any of the respective procedures.

The ability of the user to display a significant portion of the data as well as many related relationships is best exemplified in the "default output." This output is a thirteen page listing, part of which is presented in Figure 1, and consists of the following:

A. Organized Data List. The organized data list contains all 135 data items plus 13 derived items for the last ten years.

B. Ratio Analysis. Here FAS computes 30 predefined ratios.

C. Trend Analysis. FAS computes least squares trend for a selected set of items and ratios.

Moreover, the user need provide only a request card (no procedure cards) to secure this default output.

The ability to organize this data via financial statements is an important part of any financial analysis. The user may easily generate financials with the (STATEMENTS) procedure. Industry composites, which would take days to collect and compute, are easily provided by AVG or user supplied COMPUTEs, GROUPs and SCREENs. This is a further indication of the flexibility of FAS.

For the user who wishes to undertake modeling applications, REGRESSION and TREND are available. REGRESSION, for instance, provides extensive distributional analysis for each parameter. In addition, the I/O capabilities afforded by RECORD, INPUT, and OUTPUT provide very easy access by FAS to foreign data or to make available COMPUSTAT data for processing by other software systems.

EDUCATIONAL UTILITY OF FAS

In the academic community, the display, analytical modeling and I/O capabilities of FAS seem generally sufficient to make frequent and effective use of the COMPUSTAT data base. Although there is an increasing trend toward more available and up-to-date computer resources at the larger colleges and universities, this is not true in all instances. However, users of FAS would find it very desirable to have additional capabilities which include a prompting, conversational form for interactive use and extend FAS so that it could address the quarterly data files as well.

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M        REL 4.0                MOD
COPYRIGHT MAY 1978 - MSD, INC.


                               TITLE A DEFAULT LISTING OF BOISE CASCADE CORP.;
                               002400.097383
```

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M        REL 4.0                          MOD 0                                  PAGE   2
COPYRIGHT MAY 1978 - MSD, INC.          A DEFAULT LISTING OF BOISE CASCADE CORP.
INDUSTRY NAME: LUMBER & WOOD PRODUCTS            COMPANY NAME: BOISE CASCADE CORP                          TICKER SYMBOL: BCC
INDUSTRY NUMBER:    2400                         COMPANY NUMBER:   97383                                  CUSIP ISSUE NUMBER:      103
FILE CODE: PRIMARY INDUSTRIAL FILE AND S&P 425 INDEX
EXCHANGE LISTING AND S&P INDEX CODE: NYSE AND IN THE S&P 425 INDEX
*** FOOTNOTES ***       THERE ARE SPECIAL TREATMENT FOOTNOTES FOR THIS COMPANY
          YEAR: 57
FOOTNOTE NUMBER:  3

                                   O R G A N I Z E D   D A T A   L I S T
                                   =======================================


                                     *** BALANCE SHEET ITEMS - ASSETS ***

                                1              2              67             3              76             77             78
         END OF                                          RECEIVABLES .                                     WORK
         FISCAL                                          (ESTIMATED                      RAW               IN
YEAR     YEAR                 CASH         RECEIVABLES   DOUBTFUL)      INVENTORIES    MATERIALS        PROCESS        FINISHED
                              MMS             MMS          . MMS           MMS            MMS             MMS           GOODS
                                                                                                                        MMS

 67      DECEMBER            18.1000        88.0000         .0001        97.9000          .0001           .0001          .0001
 68      DECEMBER            10.5000        95.7000       . .0001       112.2000          .0001           .0001          .0001
 69      DECEMBER            90.5430       228.4330        2.4720       161.3650        85.9370           .0001          .0001
 70      DECEMBER            97.7530       219.9580        2.5660       167.3180        86.4990           .0001          .0001
 71      DECEMBER            78.9410       214.8380        5.0300       194.9280       102.3640           .0001          .0001
 72      DECEMBER            71.2670       170.1960        2.8750       180.4100        95.9560           .0001          .0001
 73      DECEMBER           186.7510       141.4060        2.8000       195.7250       103.1230           .0001          .0001
 74      DECEMBER            92.6720       129.7350        2.7810       248.6220       145.9480           .0001          .0001
 75      DECEMBER            34.0710       140.0770        2.6090       261.8520       151.4610           .0001          .0001
 76      DECEMBER           105.2860       204.6750        2.2330       313.3260       174.8370           .0001          .0001

          68                  4              7              8              73             31             32              33
        CURRENT            CURRENT                                    CONSTRUCTION   INVESTMENTS IN  INVESTMENTS IN
        ASSETS             ASSETS          PLANT-         PLANT-          IN          AND ADVANCES    AND ADVANCES
YEAR    (OTHER)            (TOTAL)         GROSS           NET       PROGRESS(NET)     TO SUBS.       TO OTHERS     INTANGIBLES
         MMS                MMS             MMS            MMS            MMS            MMS             MMS            MMS

 67      .0000            204.0000        563.9000       403.3000         .0001        11.2320         31.5730         .0000
 68      .0000            218.4000        554.4000       363.4000         .0001        48.8920         26.6970         .0000
 69      .0000            480.3410        844.2820       560.7310         .0000        81.2790        265.2600         .0000
 70      .0000            485.0290        948.0570       641.4470         .0000          .0000        300.6570         .0000
 71      .0000            488.7070       1040.7260       702.9410         .0000          .0000        271.2920         .0000
 72    83.4290            505.3020        907.4320       585.18[illegible]0         .0000        59.9660        159.1920         .0000
 73      .0000            523.8820        891.2700       560.8240         .0000        67.7850         84.3640         .0000
 74      .0000            471.0290       1016.9840       655.5680         .0000        76.8400         70.0050         .0000
 75      .0000            436.0000       1169.1870       765.4240         .0000        81.1290         58.0740         .0000
```

```
FINANCIAL ANALYSIS SYSTEM          REL 4.0                     MOD 0                    PAGE   1
COPYRIGHT MAY 1978 - MSD, INC.

                  TITLE A DEFAULT LISTING OF BOISE CASCADE CORP.;
                  002400.097383
```

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M        REL 4.0                         MOD 0
COPYRIGHT MAY 1978 - MSD, INC.          A DEFAULT LISTING OF BOISE CASCADE CORP.
INDUSTRY NAME: LUMBER & WOOD PRODUCTS          COMPANY NAME: BOISE CASCADE CORP                    TICKER SYMBOL: BCC
INDUSTRY NUMBER:   2400                        COMPANY NUMBER:  97383                               CUSIP ISSUE NUMBER:    103
FILE CODE: PRIMARY INDUSTRIAL FILE AND S&P 425 INDEX
EXCHANGE LISTING AND S&P INDEX CODE: NYSE AND IN THE S&P 425 INDEX
*** FOOTNOTES ***      THERE ARE SPECIAL TREATMENT FOOTNOTES FOR THIS COMPANY
          YEAR: 57
FOOTNOTE NUMBER:  3


                                        O R G A N I Z E D   D A T A   L I S T
                                        =====================================



                                        *** BALANCE SHEET ITEMS - ASSETS ***

                                1            2            67            3             76           77           78
           END OF                                    RECEIVABLES                                  WORK
           FISCAL                                    (ESTIMATED                       RAW          IN        FINISHED
YEAR       YEAR               CASH     RECEIVABLES   DOUBTFUL)    INVENTORIES     MATERIALS     PROCESS      GOODS
                               MMS         MMS          MMS           MMS            MMS          MMS          MMS

 67        DECEMBER        18.1000      88.0000        .0001       97.9000         .0001        .0001        .0001
 68        DECEMBER        10.5000      95.7000        .0001      112.2000         .0001        .0001        .0001
 69        DECEMBER        90.5430     228.4330       2.4720      161.3650       85.9370        .0001        .0001
 70        DECEMBER        97.7530     219.9580       2.5660      167.3180       86.4990        .0001        .0001
 71        DECEMBER        78.9410     214.8380       5.0300      194.9280      102.3640        .0001        .0001
 72        DECEMBER        71.2670     170.1960       2.8750      180.4100       95.9560        .0001        .0001
 73        DECEMBER       186.7510     141.4060       2.8000      195.7250      103.1230        .0001        .0001
 74        DECEMBER        92.6720     129.7350       2.7810      248.6220      145.9480        .0001        .0001
 75        DECEMBER        34.0710     140.0770       2.6090      261.8520      151.4610        .0001        .0001
 76        DECEMBER       105.2860     204.6750       2.2330      313.3260      174.8370        .0001        .0001

            68              4            7             8             73            31             32            33
          CURRENT        CURRENT                               CONSTRUCTION  INVESTMENTS IN  INVESTMENTS IN
          ASSETS         ASSETS       PLANT-        PLANT-          IN       AND ADVANCES   AND ADVANCES
YEAR      (OTHER)        (TOTAL)      GROSS          NET       PROGRESS(NET)   TO SUBS.       TO OTHERS     INTANGIBLES
           MMS            MMS          MMS           MMS           MMS           MMS            MMS            MMS

 67        .0000        204.0000     563.9000      403.3000        .0001       11.2320        31.5730        .0000
 68        .0000        218.4000     554.4000      363.4000        .0001       48.8920        26.6970        .0000
 69        .0000        480.3410     844.2820      560.7310        .0000       81.2790       265.2600        .0000
 70        .0000        485.0290     948.0570      641.4470        .0000         .0000       300.6570        .0000
 71        .0000        488.7070    1040.7260      702.9410        .0000         .0000       271.2920        .0000
 72      83.4290        505.3020     907.4320      585.1830        .0000       59.9660       159.1920        .0000
 73        .0000        523.8820     891.2700      560.8240        .0000       67.7850        84.3640        .0000
 74        .0000        471.0290    1016.9840      655.5680        .0000       76.8400        70.0050        .0000
 75        .0000        436.0000    1169.1870      765.4240        .0000       81.1290        58.0740        .0000
```

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M     REL 4.0                MOD 0
COPYRIGHT MAY 1978 - MSD, INC.        A DEFAULT LISTING OF BOISE CASCADE CORP.
                                      O R G A N I Z E D   R A T I O S   L I S T
                                      =========================================

                                         LIQUIDITY RATIOS
                   ------------------------------------------------------------------------------------
                         1            2               3               4              5              6
       FISCAL                        ACID                                                       SALES PER
       YEAR           CURRENT        TEST           CASH         RECEIVABLE      INVENTORY      $ WORKING
YEAR   MONTH           RATIO         RATIO        TURNOVER        TURNOVER       TURNOVER        CAPITAL

 67    DECEMBER        1.6361         .8509        42.4972          8.7409         7.8570         9.6980
 68    DECEMBER        1.6012         .7786        97.7619         10.7262         9.1488        12.5177
 69    DECEMBER        1.5021         .9975        19.0606          7.5550        10.6950        10.7480
 70    DECEMBER        1.9295        1.2639        17.5632          7.8054        10.2611         7.3479
 71    DECEMBER        1.6179         .9726        22.6228          8.3126         9.1617         9.5681
 72    DECEMBER        1.2754         .6094        16.1491          6.7622         6.3794        10.5491
 73    DECEMBER        1.6971        1.0630         7.0919          9.3661         6.7667         6.1548
 74    DECEMBER        1.6533         .7806        15.6849         11.2040         5.8464         7.8096
 75    DECEMBER        1.4336         .5726        42.7945         10.4089         5.5682        11.0570
 76    DECEMBER        2.0917        1.0402        18.3456          9.4371         6.1646         5.9377

                         ASSET RATIOS                                     EFFICIENCY RATIOS
       -------------------------------------------------------    ---------------------------------------
          7             8               9              10              11            12            13
        CASH/      RECEIVABLES/    INVENTORY/    INTANGIBLES/
        TOTAL         TOTAL           TOTAL          TOTAL           EARNING        PROFIT         SALES
YEAR    ASSETS        ASSETS          ASSETS         ASSETS           POWER         MARGIN        TURNOVER

 67     .0209         .1017           .1132          .0000           5.7854        10.8450         1.5938
 68     .0102         .0931           .1092          .0000           3.5224        12.3186         2.3046
 69     .0456         .1151           .0813          .0000           4.2986         9.7230         2.3926
 70     .0431         .0970           .0738          .0000           9.0431         5.6365         1.9619
 71     .0360         .0979           .0888          .0000         154.4425          .3225         2.0075
 72     .0400         .0955           .1012          .0000           5.0584        11.9258         1.6577
 73     .1178         .0892           .1235          .0000           3.7314        15.7025         1.7067
 74     .0588         .0823           .1578          .0000           3.9143        14.7934         1.7269
 75     .0217         .0892           .1668          .0000           5.9161        10.4022         1.6249
 76     .0608         .1182           .1810          .0000           5.4249        11.7094         1.5743
```

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M      REL 4.0                    MOD 0                              PAGE  11
COPYRIGHT MAY 1978 - MSD, INC.          A DEFAULT LISTING OF BOISE CASCADE CORP.
INDUSTRY NAME: LUMBER & WOOD PRODUCTS          COMPANY NAME: BOISE CASCADE CORP                  TICKER SYMBOL: BCC
INDUSTRY NUMBER:   2400                        COMPANY NUMBER:  97383                            CUSIP ISSUE NUMBER:     103

           25               36               40
         COMMON                         COMMON SHARES
         SHARES          RETAINED       RESERVED FOR
YEAR   OUTSTANDING       EARNINGS        CONVERSION
           M$              MM$              MM$

 75   29417.9961        409.1960        1139.9990
 76   29458.9961        487.3280        1099.0000

                                        *** MISCELLANEOUS - OTHER ***

           51               29               59               96               95               98               52
                                                        MINIMUM RENTAL   MINIMUM RENTAL                      TAX LOSS
       INVESTMENT                       INVENTORY        COMMITMENT-      COMMITMENT-        ORDER            CARRY
YEAR     CREDIT         EMPLOYEES       VALUATION          1 YEAR           5 YEARS          BACKLOG          FORWARD
          MM$               M$             CODE             MM$              MM$              MM$              MM$

 67      2.9500         27.3000           .0001            .0001            .0001            .0001            .0000
 68      1.7200         29.9000           .0001            .0001            .0001            .0001            .0000
 69      2.3000         51.9000           .0001            .0001            .0001            .0001            .0000
 70      3.0500         47.8890           .0001            .0001            .0001            .0001            .0000
 71       .6300         51.0000          4.0000            .0001            .0001            .0001            .0000
 72      1.6000         29.0000          4.0000            .0001            .0001            .0001            .0000
 73      1.7890         28.0000          4.0000            .0001            .0001            .0001            .0000
 74      3.2250         27.7110          4.0000           8.9840          31.2280            .0001            .0000
 75      7.0580         28.9770          4.0000          13.6760          54.0870            .0001            .0000
 76     14.1680         35.2280          4.0000          12.3280          48.4650            .0001            .0000
```

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M      REL 4.0                      MOD 0
COPYRIGHT MAY 1978 - MSD, INC.        A DEFAULT LISTING OF BOISE CASCADE CORP.
INDUSTRY NAME: LUMBER & WOOD PRODUCTS          COMPANY NAME: BOISE CASCADE CORP              TICKER SYMBOL: BCC
INDUSTRY NUMBER:   2400                        COMPANY NUMBER:  97383                        CUSIP ISSUE NUMBER:    103


                                       PER SHARE DATA RATIOS
      ----------------------------------------------------------------------------------------
             25             26              27              28              29             30
                                                        CASH FLOW        PRICE/       COMMON STOCK
           PRICE/        DIVIDEND        DIVIDEND          PER            CASH           HOLDER
YEAR      EARNINGS        YIELD           PAYOUT          SHARE           FLOW           EQUITY

 67       20.6250          .6109         10.4512          3.8056         10.0139          8.5828
 68       27.2584          .3823          7.7745          4.0041         15.2678         15.6083
 69       29.5543          .3226         19.4870          3.9126         18.7315         26.8567
 70       39.8707          .5405         21.1084          2.6756         16.9471         27.9915
 71      -15.6250         1.3333        -20.5021           .4305         43.5491         25.2637
 72        8.6240         1.1236          9.7072          2.6574          4.1864         19.6538
 73        4.7578          .9164          4.3247          4.5345          3.0323         24.5731
 74        2.9915         4.1619         12.4903          5.1615          2.0343         27.6864
 75       10.9375         2.5905         28.3606          4.0708          5.8036         29.2575
 76       10.1894         2.2662         23.1267          5.7536          5.8442         31.9029


                                         T R E N D   A N A L Y S I S
                                         ===========================

THE DEPENDENT VARIABLES ARE REGRESSED OVER THE FOLLOWING YEARS: 67 68 69 70 71 72 73 74 75 76

                                                                                                                T
                 ESTIMATE OF LINEAR TREND MODEL Y=A+BT          ESTIMATE OF THE ANNUAL COMPOUND GROWTH MODEL E =E (1+G)
                                                                                                             T  0
                 - - - - - - - - - - - - - - - - - -            - - - - - - - - - - - - - - - - - - - - - - - - - - - -

     DEPENDENT                        Y        CORRELATION         GROWTH         Y        CORRELATION     FIRST         LAST
     VARIABLE           SLOPE     INTERCEPT    COEFFICIENT         FACTOR     INTERCEPT    COEFFICIENT     YEAR          YEAR
        (Y)              (B)          (A)          (R)               (G)          (A)          (R)         VALUE         VALUE

EARNINGS PER SHARE      .1539       1.1194        .3433             .0001        .0001        .0001        .0001         .0001

NET PROFIT TO COMMON   7.0747      14.7961        .5070             .0001        .0001        .0001        .0001         .0001

NET SALES             62.4744    1090.6586        .5101             .0539       6.9449        .5521    1093.8603     1754.9557

DIVIDEND PER SHARE      .0481        .0616        .6885             .1139      -1.8818        .5455        .1697         .4478

AVERAGE MARKET PRICE  -4.3592      56.9004       -.6081            -.1227       4.0052       -.5679      48.1495       14.8233

BOOK VALUE PER SHARE   1.6579      14.9501        .7242             .0908       2.6507        .7057      15.4506       33.7766

CASH FLOW PER SHARE     .1610       2.8924        .3232             .0448        .9299        .1775       2.6479        3.9296

PE RATIO              -2.5580      27.9874       -.4879             .0001        .0001        .0001        .0001         .0001
```

```
F I N A N C I A L   A N A L Y S I S   S Y S T E M       REL 4.0                      MOD 0                        PAGE  13
COPYRIGHT MAY 1978 - MSD, INC.        A DEFAULT LISTING OF BOISE CASCADE CORP.
INDUSTRY NAME: LUMBER & WOOD PRODUCTS          COMPANY NAME: BOISE CASÇADE CORP               TICKER SYMBOL: BCC
INDUSTRY NUMBER:    2400                       COMPANY NUMBER:  97383                         CUSIP ISSUE NUMBER:     103


                                   PROFITABILITY RATIOS
          -----------------------------------------------------------------
            14
          RETURN            15            16           17             18
         ON COMMON                                NET INCOME/    NET INCOME/
YEAR      EQUITY        EARNINGS/     EARNINGS/       TOTAL     TOTAL INVESTED
                         ASSETS         SALES        ASSETS         CAPITAL

 67      20.5938          .0850         .0956         .0328          .0490
 68      15.2448          .1152         .1154         .0442          .0740
 69       8.8631          .0803         .0923         .0383          .0605
 70       3.9806          .0453         .0598         .0156          .0257
 71      -4.7531          .0087         .0107        -.0169         -.0286
 72       6.5494          .0734         .1137         .0226          .0354
 73      12.4200          .1181         .1413         .0569          .0828
 74      12.6636          .1190         .1290         .0658          .0908
 75       7.3933          .0751         .0808         .0407          .0538
 76      10.3289          .1073         .0961         .0562          .0708

                                   LEVERAGE RATIOS
        -------------------------------------------------------------------------------
            19              20            21           22            23             24
        LONG TERM       LONG TERM       FIXED      CASH FLOW/     WORKING     MKT. VALUE OF
          DEBT/           DEBT/         CHARGE        TOTAL       CAPITAL/       EQUITY/
YEAR  TOTAL ASSETS.   INV. CAPITAL     COVERAGE       DEBT     TOTAL ASSETS    TOTAL DEBT

 67        .3620         54.0939        3.7673        .1476        .0917         1.4784
 68        .2459         41.1890        4.5827        .2810        .0798         4.2905
 69        .2000         31.2674        4.0494        .2788        .0809         5.2214
 70        .2152         35.0361        1.9206        .1547        .1030         2.6215
 71        .2310         38.5316         .0304        .0225        .0851          .9800
 72        .2936         45.5896        3.0583        .1256        .0612          .5258
 73        .2268         32.7412        8.1275        .3323        .1357         1.0076
 74        .2028         27.8740        9.2500        .4054        .1181          .8246
 75        .2043         26.9497        5.4526        .3047        .0840         1.7683
 76        .2475         31.1216        5.7599        .3569        .1879         2.0860
```

The increased availability and usability of software systems like FAS in concert with large data bases suggests that students with potential careers in business, particularly in accounting and finance, should indeed be exposed to this type of software and data. Recall that FAS is being utilized in commercial as well as academic circles. It is unlikely that any interest in the utilization of computer based methods will follow students into their careers unless they have been academically exposed. Moreover, the ever increasing utilization of both time-sharing and computer service vendors for data analysis, modeling and evaluation suggests a corresponding activity in the educational curriculum.

FOOTNOTES

[1] COMPUSTAT is a registered trademark of Standard and Poor's Compustat Services, Inc. and is used to represent a set of proprietary data products, produced and marketed by Standard and Poor's Compustat Services, Inc.

[2] For a more extensive description, see Carl E. Ferguson and Warren G. Glimpse, Financial Analysis System - User's Guide, Management System Developments, Inc., 1978.

[3] Reproduced from the FAS manual with permission.

87

REFERENCES

Ferguson, Carl E. and Warren G. Glimpse. Financial Analysis System - A User's Guide. Management System Developments, Inc., 1978.

CAI AND THE ARTS

Warren Pepperdine
Professor, Communication Arts
Indiana University at South Bend

I have been asked to tell you how I, as a theatre man, went about learning and using the computer as a teaching device so that, I suspect, you might perceive ways of interesting others in the Arts in its use, and so help justify the expense of Computer Center programs. (An exercise we must all do from time to time to convince other administrators that we and our programs are indispensable.)

It is neither a long story nor an unusual one.

I am a three year old talking to adults. I say this not out of whimsy but because my command of computer languages and usage is most elemental. I can find food and drink and the bathroom...on occasion. My comments, then, may appear obvious and simple,though I trust, not simple minded.

I respond as an artist examining a strange new tool that might be useful in a field not usually thought of as computer territory. Colleagues in the Arts consider fasincation with the computer as just short of bizarre. What fascinates is: to what use can the computer be put to instruct students in the Arts...Students who have little mathematical skill, nor much practice in logic.

There is no doubt, even to my kind, that computer devices are to be with us from now on. In South Bend, Theatre IUSB has a new light board that is a simple programmable computer; last spring, on a lark, I went through a training course for the A. B. Dick Magnacard Word Processer. Both of these machines use many of the terms and functions of the I. U. Computer System. Thus the machines are easier to use because of a previous knowledge of Basic and Mulit-tutor languages.

We all have a responsibility to help provide our students with access to new and exciting tools in order to help their transition into the adult world. Recall Hamlet's advice: "There are more things in heaven and hell, Horatio, then are dreampt on in your philosophy."

I am trained in the word, a world of imagery. The world of figures and logic is a mist. I think and respond by intuition. Words are shifting and imprecise symbols for living things in the process of change. Therefore, it follows that words must shift, change, and evolve as time passes; there is a distrust of the concrete word or idea.

Miguel de Unamuno, the Spanish existential philosopher, felt that once a word or an idea was set down, was made concrete, was made finite, that idea was then dead. It cannot be changed and so it no longer is worth much consideration. There is something to that. In a sense you all are necro-scientists.

I am told that in large central computer centers a faculty member often can get little direct help in satisfying his curiosity about computers. I am told that the idea prevails that the center is there for those who already know how to use the machines. I am told that why should they bother to attract new users when they are already overworked. I know not about the Bloomington campus' operation. If that complaint is true there, then I count myself fortunate to be in South Bend where I have been allowed and encouraged to peek into a marvelous world. Mrs. Froehlke and Mr. Koziel have given understanding and great amounts of help. If the arrogant notion is about in the land that only those who already know how to use the computer need be worked with, then perhaps the indifference shown by many professors is justified.

I am at a campus where indifference to the computer is quite strong. In space planning meetings there is talk of reducing computer space. Challenging questions are asked as to what real value is the computer to the campus, considering the expense. At IUSB a small number of the faculty have account numbers and there is practically no use of the computer for instruction. Therefore, the conclusion must be drawn that the installation is used by a few of the faculty for their research projects and by the administrators as a hugh data processing plant. It may be a naive conclusion but if this is so, I would agree with my peers that the monies spent on such an installation could be better put into library books, day care centers, parking lots, or more secretaries.

That attitude would be foolish, of course. Like it or not, we are in a machine driven society. Machines fascinate us. We think we can use machines to lighten our work and so free us from drudgery. No one who is awake can not be so unaware as to see that not just the elect, but everyone, must become familiar with the computer. This is the place where the cost accounting argument must fail. How can we, as educators, dare say that we teach if we ignore one of the most pervasive tools in the society. From the Government to the junk mail, to our salary checks to the checkout counters we use (and are at times abused) by computer handlers. We can not ignore the computer no more than we can ignore film, recordings, or books. The computer needs to be de-mystified.

It seems to me that you, to whom computers are not a mystery, need to regain your credibility. You need to find ways to welcome those who peek through your doors. Faculty members, like students, need to be trained. A Ph.D. is no guarantee of brilliance. It is as often a sign of retreat from life into a little world of ones own make believe. We can not all be Peter Pan.

If you are so inclined then, how do you hook the fish?

With little help, but a lively curiosity, I was hooked three years ago when I started to examine the computer. It was a reasonable thought. Anything with so much potential must be able to be used in the Theatre in some way. One of the first questions asked was: "What do you want it to do?" to which the answer was: "Show me what it can do and I'll tell you if I can find anything to use it for." That is a piece of intuitive thinking.

The South Bend Computer Center personnel made it possible to use a terminal. They opened the door and let me in. (Though, I suspect, with bemusement and an inward thought that "he'll soon get bored with it and go back to his scene paints and actors.") "We are such stuff as dreams are made on."

Much time was wasted on my part and on theirs. We tend to be willing to part with money more than with our times. When was the last time we offered to spend time working in a hospital or gathering and mailing supplies to refugees? It's easier and takes less commitment on our parts to write out a check.

I floundered badly. I was not told about the several systems that could be used, nor what they did or how they could be used. DEC 10, CDC 6600/172, Weird, Loop, or whatever are still mysteries. Maybe this information was not forthcoming because they did not know how I wanted to use it. The chicken and the egg.

The first attempts were ill-conceived. There was a scheme to translate French into English. With a great flourish,a mass of French text was entered and then with a primitive knowledge of a text editor, a translation into English was made. All the "est" words were changed to "is," and all the "la" and "le" words were changed to "the." You can imagine the garbage. Translated imbedded strings make curious reading.

Even though they knew little about it, the Center's managers pointed to the Multi-tutor language and to two manuals: The Northwestern manual on "Plato" and B. A. Sherwood's "The Tutor Language." These manuals gave enough technical information to go ahead with a beginning project: a study-outline for the Classical Athenian Theatre. Those are the only two works I know of on Multi-tutor. They apply only in part to the I. U. system. We have nothing that is meant just for our system and, more seriously in my view, no material that has been prepared for those who have no knowledge of computers and mathematics.

The first project, that on the Classical Athenian Theatre, took nearly a year of off-and-on work to perfect. No one in South Bend seemed to know anything about Multi-tutor, and Joe Uthuppuru was too far south in Bloomington to be of much practical help.

With patience and determination, most errors were finally eliminated and a workable lesson finished.

This success led to a grant to prepare study material and questions for a History of Costume course. The proposal was to create twenty lessons, based on the text used in the course; to provide descriptions of garments worn from the Egyptians to the 20th Century; and to use reviews and test questions which were to be run by the students as drill material. The exam questions would be taken from these questions. If the lessons were used conscientiously, then it followed that test scores were bound to improve.

The grant monies were used to pay an assistant, James Conley, to enter the text, correct the errors, and to create sections of the lessons where I could not. These included sections on the adding of errors, keeping the grades, and such like feed-back items that would help to evaluate the lessons.

Our immediate problem was that the text, having been written in longhand and then typed in at the terminal, took an inordinate amount of time, plus producing too many errors.

The answer to that problem was for me to enter the material directly and to let Conley change the errors and make the necessary mathematical entries. The first lessons took two to three weeks to complete. The last several lessons were completed, in nearly finished form, in three to four hours, if the system was in good working order.

It will be no surprise to you that the most frustrating things for the instructor, as well as the students, are malfunction of the equipment and the slow response time in running the lessons. This supports the evaluation presented by Burns and Bycior during the fourth conference of this group.

By if you want to extend the use of the computer for instruction, there seems to be several possibilities. The system needs to be made even easier to use. We like to use those things that are most familiar to us. An axiom of Mass Communication is that the easier a medium is to use, the more people will use it.

1. A manual could be prepared by a computer technician and someone in language arts. This would essentially be a translation into English of technical terms and jargon of mathematics and computer language. For instance, why cannot "interface" be "link," or "hardcopy" by "copy." Also helpful would be the plentiful examples of sample problems that could be adapted for most any need. Examples: a general error counting and reporting program; or crucial parts of a whole program could be presented so that one could see how a complete program might look; a grade in program which would only need the names and grades entered.

2. A programmer at each remote center charged with assembling and maintaining the lessons may not be a pleasant thought to you. Do not dismiss it out of hand. Though all your personnel are overworked, if you don't spend a good deal of time getting support from a large part of your faculty, their indifference could work against you when buildings are raised and tight budgets are reallocated. This programmer might prepare lessons from written lectures, notes and tests provided by the instructors. This programmer needs to be able to communicate in English, to express instructions and ideas in terms devoid of jargon.

I have heard, or rather overheard, remarks that "the administration is not deeply committed to the Computer Center." or, "The Computer Center is not getting enough support from the administration." We all feel that, in whatever area we are in. But there is only so much money to be divided among those of us clamoring at the administration's doors. The complaint may be true. But is also may be true that there has not been enough effort to prove that the computer is of great value to a larger part of the faculty. More support will be available if empirical proof is given that more people can use it. Never mind the Cassandras and the pseudo-sophisticates, (or even worse, the reverse-pseudo-sophisticates) who nay-say the computer. They are afraid of it and are therefore defensive.

Could not a major effort be begun (perhaps through the local campus computer committees) that would be aimed at non-scientists. For instance, instead of sending out memos or announcements to everyone that such-and-such will be demonstrated for interested parties, why not prepare four or five lessons for History, Shakespeare, English, French, etc. These could be general system-wide lessons. They might be developed with specific professors. Then arrange to use these lessons with individual classes at all lcoations. Timing is important. Several years ago, "Plato" was demonstrated in South Bend. It came at the worst possible time, as I recall...either at mid-semester or near finals. Practically no one saw it because of that, and because the general announcement type invitation. I suspect the majority who showed up to use the terminal were already working with existing equipment.

What would happen do you suppose, with a three day festival on Computer Assisted Instruction that was properly promoted and planned. By that I mean with banners, treats, and quite a bit of showmanship. This idea includes finding the showmen who can talk to the audience in terms they can understand. This Festival should happen early in the term to envolve most of the students and faculty before their other projects claim the priority of attention.

I have gone on at some length giving you some observations made over a three year span of time about the problems and frustrations of a part of your constituancy that is not being served.

I think only with some kind of major de-mystifying and a well planned and well executed program of this kind will you be able to extend computer usage to a larger base. Then perhaps, the computer would not be thought of as a large calculator, or a machine that keeps student records, or a beast that spews out yards on controvertible figures that everyone uses to prove everyone else wrong.

It could be that the system's Computer Assisted Instructional Committee needs to press for more "prime time," to use some jargon from my own field. Where is CAI going and how will it get there?

"Plato" is indeed an attractive answer but can all the remote sights afford it? I'm told not. If that is so, how can Multi-tutor be made more effective and made into a tool seriously considered and seriously used.

THE COMPUTER AS PUBLISHING TOOL: THE IDS SYSTEM

George Alexander
Systems Co-ordinator, IDS
Indiana University

The IDS and the School of Journalism share a twin-CPU 16-terminal computer system for editing and typesetting. Because it was designed specifically for publishing, it has special software with sophisticated editing facilities, and special peripherals including an optical scanner and two phototypesetting machines. Redundancy has been provided throughout the system to try to ensure that equipment failure will not cause missed deadlines or even failure to publish. Because the system has a magnetic tape drive, data can be exchanged with other computers via magnetic tape. This capability makes possible many interesting projects, a few of which will be described.

TIDY

John H. Dustman
Associate Professor
Indiana University Northwest

In theory at least, the Indiana University Computing Network should operate with a maximum amount of compatibility between the respective main-line computers. The main-line computers are the DEC 10 system in Indianapolis, the WCC system in Bloomington, and the Data Systems and Services system in Bloomington. Each of these systems have the capability to communicate between themselves giving the individual user great freedom in transfering data and programs from one computer system to another. When established the primary concern was to insure compatible character sets and this is accomplished through HASP and IMPORT. The DEC 10 system has been extremely responsive to the needs of individual programmers and users by establishing a variety of network routines such as NPRINT, NETEXE, and NETPIP which allows printing, executing of programs, and transfering of files between the DEC 10 and WCC. One area, however, that has received little attention has been to minimize the differences between the various compilers and assemblers. Hence, BASIC on WCC is unique in many features and will not assemble properly on the DEC 10 and vice versa. Similarly, FORTRAN programs written on either of the two machines will not compile without errors on the other. In particular the handling of IO differs between the two machines. FORTRAN programmers and users are consequently faced with the problem that programs written for one machine will not run on the other without substantial changes. Ideally, some method should be developed to minimize the changes needed to be made by an individual to allow his or her program to run on both machines. The development of the TIDY system on the DEC 10 is an attempt to do this.

Those programmers who have used the WCC system extensively are probably aware of a library routine called TIDY which operates on the CDC 6600. It was developed primarily to renumber, edit, and in general clean up FORTRAN source programs whose statement numbering has become unwieldy and whose readability has deteriorated as a result of many revisions, patches, and corrections. Such programs frequently result when several programmers work together on the development of a very complex code. WCC TIDY will produce a new source deck which has been cleaned and reordered. Using this program and its features as a base, I have developed a DEC 10 program called TIDY which accomplished the same features. However, in addition allows the user to determine whether the new source code should be in DEC 10 style or WCC style FORTRAN code. It, therefore, can function as a translator between the two systems to maximize program interchangeability.

DEC 10 TIDY provides a simple means for "clearing" up the appearance of FORTRAN programs. TIDY reads the old FORTRAN program routine-by-routine, and prepares a new version of the program which has the following characteristics:

1. All statement numbers are right justified and increase in consecutive order. The increment the numbers increase by can be set by the user.

2. Statement numbers are assigned only to statements that are referenced by other statements. Unused statement numbers are deleted from the final source code.

3. All format statements are collected from within the body of the program and placed at the end. Format statement numbers are edit and resequenced in the final source code.

4. All excessive blanks are deleted from each statement, while blanks are inserted as necessary to ensure uniformity and to improve readability.

5. All statements begin in column 7 unless they are nested within a DO loop. DO loops are indented as they are nested and the number of columns used to indent the statements within a DO loop is set by the user.

6. All comment cards begin their text material in column 7 or can be centered within the card image. All blank comments beyond 1 are deleted.

7. All statements in each new routine are labeled in card columns 73 through 79 with a unique letter-number combination. If routines are to be retained on the DEC 10 it is suggested to request TIDY to remove the card sequence numbers to allow for minimum block storage requirements. When sequence numbers are removed the term DATA or FORMAT are retained in columns 73 through 79 for data and format statements to protect possible hollerith fields that might be destroyed when trailing blanks are truncated.

8. Multiple statements separated by $ for WCC originated programs or ; for DEC originated programs are separated into individual statements.

9. When DEC 10 FORTRAN code is translated into WCC FORTRAN all statements unique to the DEC 10 are either translated to their WCC counterpart, converted to comments, or individually noted.

10. Although WCC-TIDY would not accept implicit end= or err= parameters in read statements, DEC-TIDY allows these statements.

11. Most FORTRAN II statements are rewritten into FORTRAN IV statements. The key word TYPE in type declaration statements is deleted. For example, "Type Integer ABC" will be changed to "Integer ABC." However, the TYPE and ACCEPT IO statements used by DEC-FORTRAN will be retained or converted to appropriate read and write statements depending upon whether the user translates DEC programs into WCC programs. Print statements can either be retained or converted to a write statement.

12. Read and write unit numbers can be converted or changed as desired.

13. All variable names above 6 characters can be truncated to a standard 6 character name which is unique or individual variable names can be renamed to specified alternative names.

To make the program as compatible as possible most of the standard WCC-TIDY control codes were retained unchanged and the user can use "CALL,WRITEUP(N=TIDY)" to obtain a full documentation on the WCC-TIDY system. In addition, when running the DEC-TIDY program all control codes are listed and an explanation of each can be obtained at run time by requesting "HELP."

The following are specific instructions on how to use the DEC-TIDY system. The TIDY system exists on PPN 57001,1000 as a file called "TIDY.SAV". To run the program enter the following command:

```
"RUN TIDY(57001,1000)"
```

The program will respond by requesting an input and output file name. The input file is the program you wish to be cleaned or translated, and the output file name is the name you wish to give the resulting deck. Although the same file name can be used for both this is not recommended unless you are sure no errors or problems exist in the source deck. The program then requests whether the user wants the new file in DEC or WCC style FORTRAN. If WCC style FORTRAN is requested the program will request IO unit numbers for converting TYPE and ACCEPT to WRITE and READ statements. The program then will list the control codes or statements and allow the user to obtain detailed descriptions of each command. Following this the user enters which ever commands he desires. The suggested commands are:

Base=10.	Set the base number to 10 for beginning statement numbers
Statement Increment=10.	Increment statement numbers by 10
Indent-5.	Indent DO loops 5 spaces
References	Produce a reference list of all old and new statement numbers.

Entering a blank link terminates control card input and begins the editing of the program. Progress statements are printed periodically to indicate which routine TIDY is currently working on. Following the last routine, the program will request whether or not sequence numbers in COLS 73 to 80 should be removed or kept. After the program finishes, a file called TIDY.LPT will contain a listing of the old and new program statements and should be sent to the printer with a FILE:FORTRAN switch. Trailing blanks can be removed by using PIP with a T switch.

Bugs, problems, and other little things: to date very few "bugs" have been discovered. The program does screw up when you have a read statement with a variable format and either an implicit end= or err= branch in the statement. The program does not check for unique library subroutines which may not be duplicated in both machines. This checking must be done individually. Core problems may develop if the original program is above 600 statements in length since various intermediate files are produced and the user may find himself exceeding core limits for his PPN. The only alternative in this situation is to process the routines individually on in small groups. Random access read and write statements are accepted by the program and are not translated into WCC equivalent statements. The user must substitute appropriate WCC routines such as described in RIOLIB. WCC translated programs will also need the addition of a file equivalence card at the beginning of the program. This card is not produced by TIDY.

In summary, the TIDY program on the DEC 10 accomplishes most of the features of its WCC counterpart and in addition functions as a FORTRAN translator which should allow for much easier conversion of programs between the DEC 10 and WCC or D. S. & S. With increased program interchange the I. U. Computing Network should be able to take one more step towards its goal of being a true functional network.

EXAMS: COMPUTER GENERATED EXAMINATION SYSTEM

John H. Dustman
Associate Professor
Indiana University Northwest

The preparation and grading examinations is one function of teaching each of us must accomplish on a regular basis, and one which can take a considerable amount of time and effort. Last year I presented a paper which detailed a series of programs which graded examinations and maintained a grade record for each student within a course. The next logical step appeared to be the development of a system which could generate examination questions and prepare individual class examinations which could then be easily graded by this system. Program EXAMS is a program which allows the user to establish files of multiple choice type questions which can be used to generate examinations quickly and efficiently. The major features of this system are as follows:

1. Individual files of questions can be generated by subject area with no limit on the number of questions within each file. Each question can have up to 200 characters and up to five different answers each containing 100 characters of information.

2. Question files may be listed in their entirety or partially. Each question is listed along with the answer choices and a record of how many students have attempted the question and how many have had it correct (difficulty factor). When questions are listed, the A) answer is always the correct response, followed by the incorrect answer choices.

3. Each time an examination is requested, the answer choices for each question are randomized. However, if "true" and "false" are the answer choices they will always occur in that order. "All of the above" and "none of the above" will always appear in their respective positions.

4. All question files are stored in binary records to avoid the possibility of a student simply listing off a question file. Any attempt to type or print such a file will result in only one line of print which only contains the file name and course title information. However, using a "dump" feature, any question file can be converted to a 80 column card file.

5. When an examination is produced, the output will be in both upper-case and lower-case format and will be spaced and indented appropriately. The program assumes that the first character of the question and each of the answers should be in upper-case. To have any other character in upper case preceed that character with the code when the file question is initially made. There are two formats used to output the examinations. They are: short and long. An example of the "short" format is:

   1. The largest organ of the human body is A) liver B) nose C) skin D) tongue E) none of the above

2. The hormone ADH functions to A) favor reabsorption of Na by the kidney B) promote reabsorption of the water by the small intestine C) promote reabsorption of water by the collecting tubules D) all of the above E) none of the above.

An example of the "long" format is:

1. The largest organ of the human body is

   A) liver
   B) nose
   C) tongue
   D) skin
   E) none of the above

2. The hormone ADH functions to

   A) Promote reabsorption of water by the collecting of tubules
   B) Promote reabsorption of water by the small intestine
   C) Favor reabsorption of Na by the kidney
   D) All of the above
   E) None of the above

The "short" format is best for those who wish to save paper, whereas the "long" format is the most readable.

6. When examinations are prepared course title, instructor, and date will be entered into the first page of the examination along with any directions desired by the instructor.

7. All questions in the files may be edited or changed as necessary by using the edit command.

8. Examination output and file listing are post-processed by two MIC files which converts the raw output files to the finished copies. The finished copies can be typed on any hard copy terminal and one copy is automatically sent to the printer. In addition, by substituting either ditto or memeograph stencils for the paper in the hard copy terminal, direct masters can be produced, thus eliminating the need for a secretary to type the examination.

9. When examinations are made, two additional information files are produced. One is a file called KEY.DAT which contains a list of the question numbers and the appropriate answer as follows:

```
1. D) All of the above
2. A) Approximately 6 inches
3. E) Estrogen and progesterone decrease markedly
```

A second file is produced called VALUE.DAT. This file contains a list of the questions used according to file or origin and difficulty factor. The output appears as follows:

```
234  REPRO.DAT    45.67
 12  REPRO.DAT    88.90
134  URINE.DAT    78.88
 99  SKIN.DAT     ******   (File question not used previously)
```

When the examination is graded using the Indiana University Northwest Grading System, this file is used to keep a record of the number of students getting each question correct and they can be used to update the original question files with the new information gained from this particular examination.

10. Examinations can be produced where the user either selects the questions desired from individual files or where the program will randomly select a specified number of questions from each file desired. Each question selected has a previous difficulty rating if used at lease once and at the completion of preparing an examination the average difficulty for the examination is calculated.

In order to run the program the user needs to enter the following command:

```
"RUN EXAMS,SAV(57001,1000)"
```

The program will respond by listing the commands available to the user. The commands and their functions are as follows:

ADD	To add questions to an existing file
MAKE	To prepare an examination
EDIT	To edit a question on an existing file
VALUE	To update the difficulty factors for specific questions
LIST	To produce a listing of questions contained within a file
DUMP	To dump a binary question file to card image
RESTORE	To restore a card image file to a binary question file
END	To end the run and finish the program

After entering one of the above commands (except end) the program will request a file name and extension. If the file does not exist it will either make a new file (in the add function) or will request the PPN number the file is located on. This allows storage of files on associated PPN numbers.

The add function allows you to add additional questions to the end of an existing file. By convention, it is suggested that the first question entry in each file contain the course title and instructor, and file name as the question and first answer choice. The program will never use question one of any file as an examination question.

The EDIT function allows the user to change individual questions or parts of questions. Each entry is listed beginning with the question and then each of the answer choices. A blank line assumes the contents to be correct, any entry of new text material will replace the old contents of that line, and an * followed by 4 spaces means blank out that entry.

The MAKE function allows the user to prepare an examination. It will request course title, instructor, date, and directions of comments. It will also request whether a "short" or "long" format is to be produced and whether questions are to be randomly or individually selected by the user. Questions are requested by question file. When no additional questions are desired from a particular file a blank entry will transfer to the request whether an additonal file is to be used. If an additional question file is to be used it will allow the user to choose between random or individual modes of selection. If no additional files or questions are to be used the program returns to the function statement.

The LIST function allows the listing of the entire question file or selective parts. If partial listing is desired a starting and stopping question number are requested. If no end number is given it will assume that the last question is the terminating question. It will also request a page number to label the output so that sequential pages can be substituted with ease.

The DUMP function produces an 80 character card image deck of question file with appropriate information about difficulty factor and usage. This file then can be transferred to WCC for use on that machine. A second version of EXAMS which will run on WCC is near completion and the card image decks will allow the complete transferring of files between the two machines. The RESTORE functions reads the card image deck and produces a binary question file. The DUMP/RESTORE aspect will be particularly useful for those individuals who wish to maintain files on both operating systems.

The VALUE function allows either selectively updating question files for difficulty levels on questions used or using the output generated by the examination grading system to update the question files.

In summary, program EXAMS affords a easy and efficient mechanism to generate and maintain examination question files, prepare class examinations, and evaluate effectiveness of questions. Assuming question files have already been established, a 100 question examination requires approximately 5 minutes to generate the examination file, and another 10 minutes to list it for duplicating. In total, it requires approximately 200 kilocore seconds or about $2.00 in computer time.

MIST: A Music Information System

Rosalee Nerheim
Wrubel Computing Center
Indiana University, Bloomington, IN 47401

Information systems have proved their usefulness in many disciplines. Until now, they have had very limited success in music mainly because music differs so radically from other types of data. MIST was written to deal with this problem.

A music information system can deal with two types of music data. One type is bibliographic data about music and music literature; the other is the music itself. This second type of data poses far greater problems. Music must be converted to some machine readable form, usually alphanumeric. This is costly and inconvenient because music notation, a two dimensional system, differs so radically from alphanumeric code, a one dimensional system. More problems arise when a scholar just wants an excerpt from a file of music data. This sort of searching becomes very difficult when working with any large corpus of data. For this reason, most operational systems use incipits only which limits their applicability.

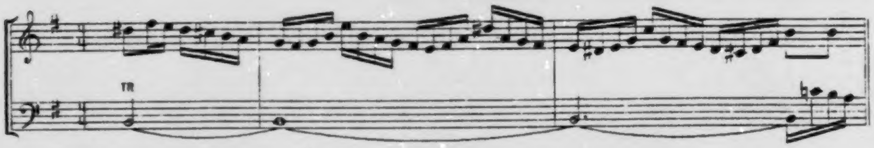
The Music Information System for Theorists (MIST) was written by the author to help music scholars with some of their data handling problems. With MIST a scholar can retrieve pedagogical excerpts by specifying attributes of composer, date, style, and/or desired concepts to be exhibited by the excerpt, or he can search for entire works by composer. The user types his requests at a terminal and a list of excerpts or works satisfying his request is returned to him. He may choose to print the music data in conventional music notation or he can get a copy of the data in alphanumeric form for further work. MIST is flexible in that its data base can be enlarged and corrected as needed.

MIST consists of three programs. These programs tie into the Indiana University Computer Music System, which can print the music, analyze various aspects of the music, or realize it in sound. INTRACT is an interactive program that allows a user to search for music in the ways already mentioned. When a user requests some music data, INTRACT saves this information and passes it to REQUEST. REQUEST retrieves the desired music data and directs the I. U. Computer Music System to perform further processing. The third program, UPDATE/INSPECT, edits the data base. The user need only be aware of INTRACT and the final product.

MIST Record Types

MIST's data base uses two record types. One type contains the music itself, the other contains bibliographic information. The former always contains a complete work. MIST's alphanumeric representation of music is MUSTRAN. A translator program converts the MUSTRAN to its internal form, which has an elaborately linked structure. When each work of music is added to the data base, it is stored with a work identification number along with its MUSTRAN internal form on any storage device.

The other record type, containing the bibliographic information, requires a random access device. Each bibliographic record holds information describing an excerpt or complete work. A record describing an excerpt includes composer, title, concept(s) exhibited, and a pointer to where the excerpt occurs in the MUSTRAN data. The pointer consists of work identification number, voice(s), inclusive measure numbers, and starting beat. Consider the following example of a pedal.



The record describing this passage contains the following data:

J. S. BACH	(composer)
TWO PART INVENTION NO. 7 MEAS. 15-17	(title)
1723	(date)
PEDAL	(concept exhibited)
W16.V1,2.M15-17.B3.	(pointer)

This pointer indicates that this excerpt can be found in work number 16, voices one and two, measures 15-17, starting on beat three of measure 15.

The program REQUEST can take this pointer information, pull the proper work from the music data, and use the link structure of the MUSTRAN internal form to find the proper voice and measures. Traversing the links is fast and cheap.

The bibliographic record type is also used to encode data about a complete work. Note that many bibliographic records, one describing the complete work and the rest describing excerpts, can point to a single music record.

All information about excerpts is gathered using traditional (i.e., manual) analysis. MIST itself does no analysis.

MIST searching commands

INTRACT, the searching program, runs interactively. A user can ask for a list of music excerpts via the FIND command. This is the most important command. Specifications for FIND are given by the criteria COMPOSER, CONCEPT and DATE.

COMPOSER narrows the search to works by a particular composer. To look for anything written by Beethoven, the user types:

```
COMPOSER = BEETHOVEN
```

CONCEPT is the criterion which is matched against the concept information in the bibliographic records. To search for examples of hocket, the user would type:

```
CONCEPT = HOCKET
```

Up to three concepts can be specified in one search.

DATE gives the desired range of dates of composition.

```
DATE = 1685-1750
```

Other commands include COMPOSERS (list composers in data base); CONCEPTS (list concepts); PRINT,nnn (print excerpt/work nnn in conventional music notation); DATA,nnn (output excerpt/work nnn in MUSTRAN source and HELP.

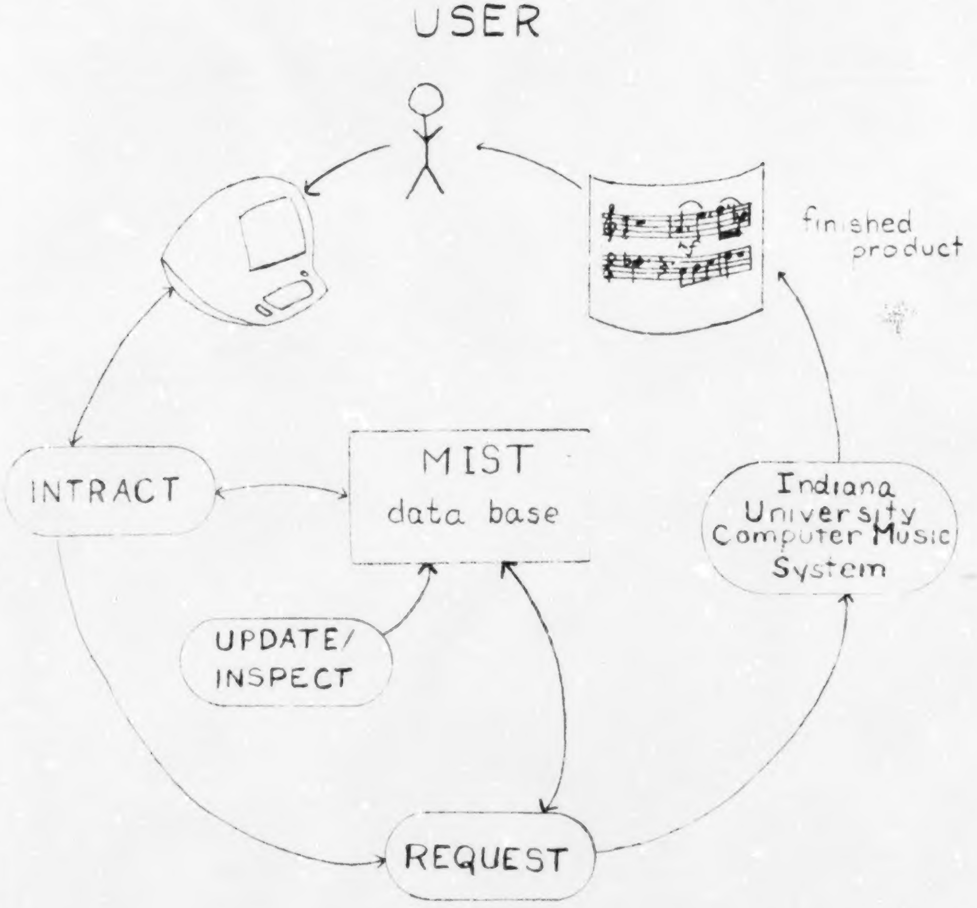

USER
finished product
INTRACT
MIST data base
Indiana University Computer Music System
UPDATE/ INSPECT
REQUEST

Figure 1: MIST programs, the MIST data base, and the Indiana University Computer Music System

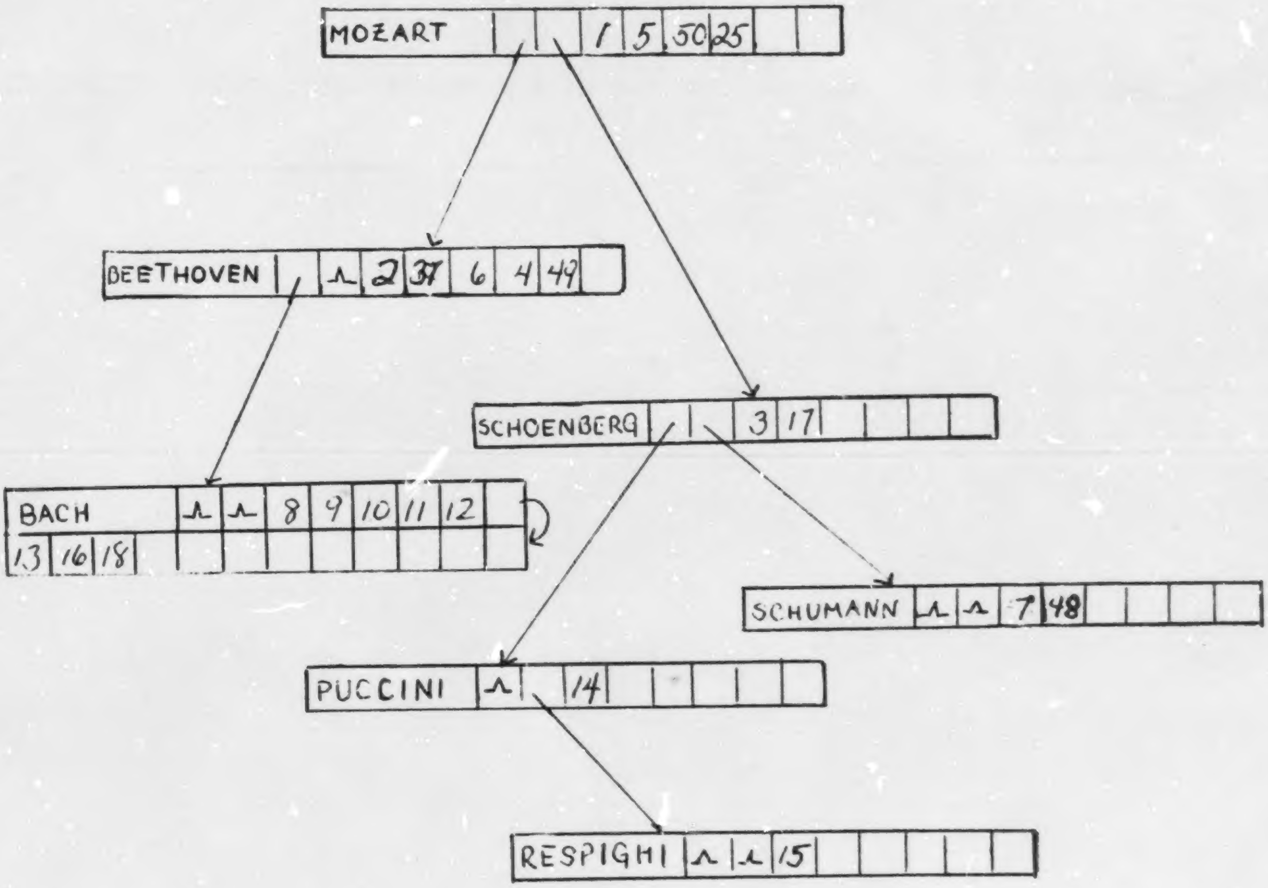

MOZART
1
5
50
25
BEETHOVEN
2
37
6
4
49
SCHOENBERG
3
17
BACH
8
9
10
11
12
13
16
18
SCHUMANN
7
48
PUCCINI
14
RESPIGHI
15

Figure 2: Search tree used by MIST

USING AN EXPER SIM (EXPERIMENTAL SIMULATION) MODEL IN TEACHING GRADUATE RESEARCH COURSES IN MUSIC EDUCATION

Hal Abeles
Associate Professor of Music
Indiana University

EXPER SIM (Program: Michigan Experimental Simulation Supervision-MESS) is a computer-based experimental simulation system which allows instructors to develop models of experiments in their disciplines as well as providing access to simulations developed by other model builders. Several EXPER SIM models have been developed for experimental simulations in disciplines such as psychology, sociology and education. The current project involves the development of two models for use in graduate research courses in music. The two experimental simulation models being developed involve research in the areas of "Experimental Aesthetics" and "Music as a Reinforcer for Non-Music Behaviors."

The experimental simulations are used to provide the students with an integrated exposure to the steps of the scientific method, without the time consuming administration problems which often accompany real data collection. The student first becomes acquainted with research in the simulation field by completing two abstracts of related research articles. Additional background information describing relevant theories and a list of additional references is provided in the Student Handbook. The students also receive a list of variables available for manipulation and available dependent variables. After the students study the background information and discuss possible relationships which may exist between the manipulable variables and the dependent variables with the instructor, students generate hypotheses and design an experiment to test them. The output provided by EXPER SIM is then analyzed employing SPSS statistical analysis programs. Students receive consultation to assist them in selecting appropriate analysis procedures. Students then submit brief reports of their projects in a journal format. Results of the projects are discussed by the class. Students are then expected to design additional studies based upon the new knowledge acquired from the first series of reports. Data analyses and reports are again required. During the semester the student submits a minimum of three research reports to the class and a final report which summarizes his as well as the work of the other students in the class.

Initial analysis of the student experiences with EXPER SIM suggest they have a better understanding of the relationship of the steps in the research process than in previous semesters. Students also have indicated strong positive attitudes (e.g. doing as many as 12 studies when only 3 were required) toward this instructional mode.

Additional information on EXPER SIM can be found in Pipeline, Vol. 3, No. 3, Summer 1978, published by CONDUIT which distributes the MESS software and documentation currently being employed at IUB.

AN ON-LINE REGISTRATION SYSTEM FOR A MINICOMPUTER VIA EVENT-DRIVEN FORTRAN

Ronald A. Barry
Dept. of Statistics & Computer Science
West Virginia University

Malcolm G. Lane
Associate Professor
West Virginia University

INTRODUCTION

Many small institutions have insufficient in-house computing power which generally tends toward a batching environment. Commonly found in such facilities are minicomputers with limited memory, operating systems which do not support multitasking and application languages not compatible with the multi-terminal environment. The "80 card column syndrome" still exists in such places and has made the on-line environment look like an insurmountable task. One such institution faced with this problem is Parkersburg Community College (PCC) in Parkersburg, West Virginia -- a two year college offering baccalaureate and technical courses as well as various community services. Their computer system consists of a Digital Equipment Corporation PDP 11/40 minicomputer. The system configuration is shown in Figure 1. The main frame consists of 32,000 words of memory and a console terminal used for system/user interaction. The peripherals on the system are the card reader, line printer, two disk drives, and a magnetic tape drive. The operating system used is RT-11 version 2C. Included in its software is FORTRAN, MACRO assembler, MU-BASIC and various system utilities. In addition, a 16 port multiplexor was added to support four crts and four hardcopy terminals. However, the operating system does not support this multiplexor. Also provided with the system is batching capabilities to a dual processing system consisting of an IBM 360/75 and an ITEL AS/5. The two machines are the host computers for a state-wide network, WVNET (West Virginia Network for Educational Telecomputing). All together, the configuration shown in Figure 1 makes for a very powerful batch-oriented system.

PCC, like many other colleges, is faced with the problem of generating various required reports on all facets of the college's business including registration information. The most common type of registration system is the batch-oriented system which requires a great deal of "paper shuffling". This is not only time consuming but also, at times, inaccurate. There is an inherent time lag produced by this traditional approach. For example, business processing during registration requires quick and accurate processing by the registrar. Any time delay along the "paper flow" route will cause complications down the line.

In August, 1976, a project to develop an on-line registration system for PCC was undertaken. The idea was to provide an efficient and accurate means of registering a student for classes, maintaining biographic and demographic information on that student and to provide an easy means of billing. Current information on all courses offered at the college and and all additional reports was to be maintained on the system.

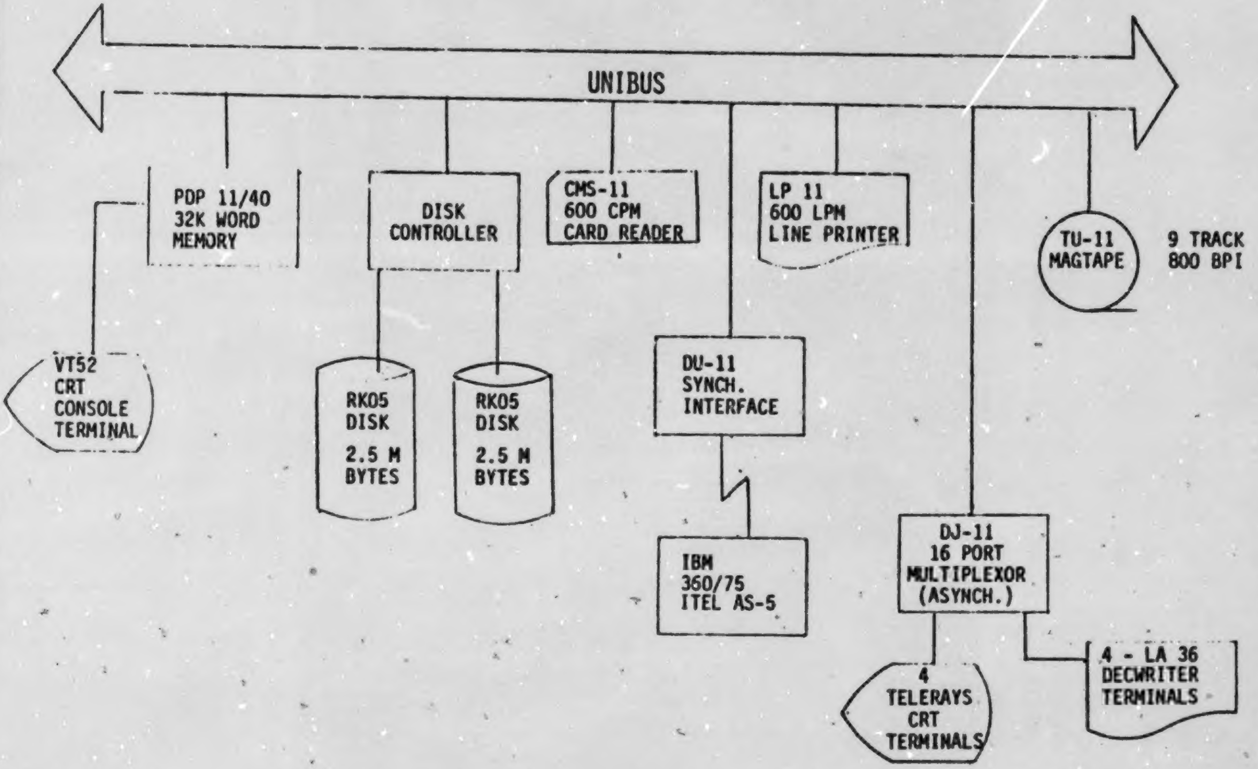

UNIBUS
PDP 11/40
32K WORD
MEMORY
DISK
CONTROLLER
CMS-11
600 CPM
CARD READER
LP 11
600 LPM
LINE PRINTER
TU-11
MAGTAPE
9 TRACK
800 BPI
VT52
CRT
CONSOLE
TERMINAL
RK05
DISK
2.5 M
BYTES
RK05
DISK
2.5 M
BYTES
DU-11
SYNCH.
INTERFACE
IBM
360/75
ITEL AS-5
DJ-11
16 PORT
MULTIPLEXOR
(ASYNCH.)
4
TELERAYS
CRT
TERMINALS
4 - LA 36
DECWRITER
TERMINALS

Figure #1 PDP 11/40 System Configuration

This paper looks, in detail, at the development and design of the On-line Registration System (ORS) for Parkersburg Community College. The system was completed in August 1977 and has been operational since that time.

DEVELOPMENT OF THE ORS SYSTEM

Initial talks with all administrators involved with registration brought out the need for a more efficient and accurate system. The system was designed to meet the following objectives:

1. Provide a means for displaying, entering and updating all biographical and demographical information on students admitted to PCC.
2. Provide a means for displaying, creating, adding, dropping and modifying students' course schedules.
3. Provide a means for displaying and clearing all business transactions created during registration.
4. Provide a means for displaying and modifying course information.
5. Provide a method of securing the data as to ensure its integrity and its mutual exclusion among the various applications.

It was also required for ORS to maintain information on approximately 6,000 students and some 1,500 courses. To achieve such objectives, it was necessary to provide for multi-terminal and multiprogramming support. Each terminal had to be physically, as well as logically, connected to the DJ-11 multiplexor and the ORS system respectfully. This then provided ORS the necessary information as to the type of data that could be sent to the terminal as well as the terminal's location. To provide for a multiprogramming environment, each terminal had to be able to execute its own application programs totally independent of any other programs being executed on other terminals. Furthermore, the control over which program to run had to be given to the user. To maintain this programming independence, the code for each application program had to be reentrant or at least pseudo-reentrant and each terminal had to be interrupt-driven.

After looking at the system's configuration (no upgrading in the near future was expected) and the objectives as stated above, the choice of the language to be used for ORS became academic. Constraints of the hardware and software on the PDP 11 posed problems which influenced the decision. For example, the amount of memory as indicated in Figure 1 is rather deceiving. The top 4,000 words are unusable without memory management facilities. PCC's system does not have this hardware option. In addition, this option is not supported in version 2C. Therefore the amount of memory available to ORS is immediately cut down to 28,000 words. The systems overhead, including the Resident Monitor (RMON) and the User Service Routine (USR), amounts to approximately 4,000 to 5,000 words. A diagram of the memory layout of the RT11 operating system is shown in Figure 2. Ultimately, the amount of memory left to ORS, providing no foreground job is present, amounts to approximately 22,000 words.

RT-11 MEMORY LAYOUT

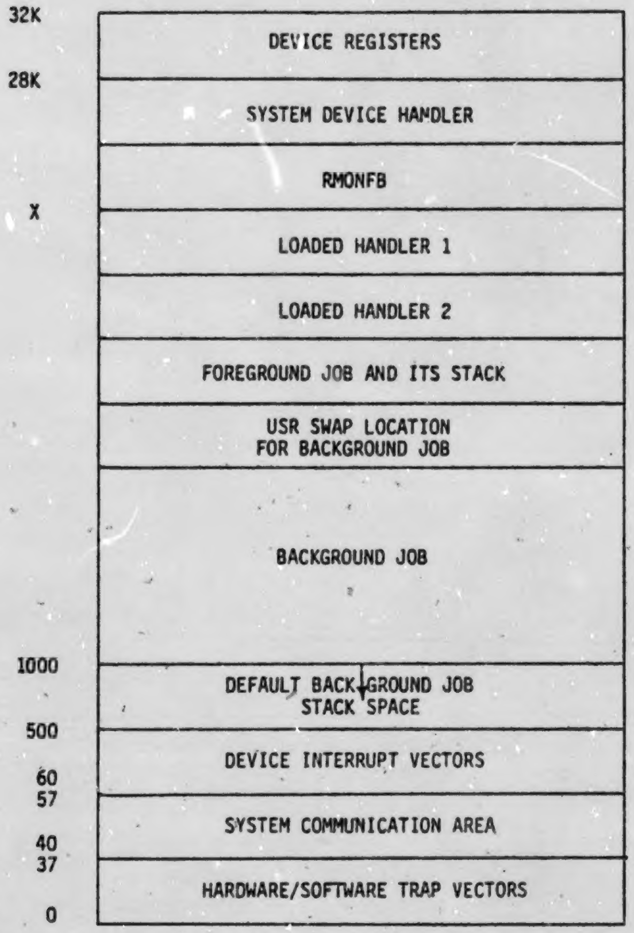

32K
DEVICE REGISTERS
28K
SYSTEM DEVICE HANDLER
RMONFB
X
LOADED HANDLER 1
LOADED HANDLER 2
FOREGROUND JOB AND ITS STACK
USR SWAP LOCATION FOR BACKGROUND JOB
BACKGROUND JOB
1000
DEFAULT BACKGROUND JOB STACK SPACE
500
DEVICE INTERRUPT VECTORS
60
57
SYSTEM COMMUNICATION AREA
40
37
HARDWARE/SOFTWARE TRAP VECTORS
0

Figure #2 FB with Loaded Device Handlers and Foreground and Background Jobs

The MU-BASIC (multi-user BASIC) software provided with RT-11 does give multi-terminal and multiprogramming support but has several drawbacks. The overhead of the interpreter alone amounts to some 10,000 words of memory. Each user is then allocated a fixed or variable amount of what is remaining. The amount assigned can not be dynamically changed without reconfiguring the system which is done prior to its execution. MU-BASIC also does not allow dynamic reconfiguration of terminal ports. Therefore, if a failure occurs on a line or a terminal, the system would have to be terminated before reconfiguration. Although MU-BASIC does support multiprogramming, it does not share the code. If two users on separate terminals wish to run the same application program, two copies of the code must be resident in memory. Despite the fact that the code need not be reentrant (a programming "relief"), memory utilization is very poor. One additional drawback is MU-BASIC's relative slowness. Since it is an interpreter, execution is done on a line per line basis only after the basic instruction has been translated. Furthermore, the chaining feature available in MU-BASIC, where software is overlayed into the user's work area, is very time consuming.

The MACRO assembler provided in RT-11 has a very powerful instruction set, is fast and efficient, and conserves on memory usage. The code can be reentrant as well as relocatable. The major concern with using this language for ORS was the large development time and the complexity of the language.

FORTRAN, although not without its faults, was the language chosen for ORS. One of the major problems with FORTRAN is its lack of character string manipulation facilities. Those facilities, though, are present in the RT-11 system subroutine library and are callable from FORTRAN. In addition to the use of the standard top-down structured programming approach, it was necessary to simulate the multi-terminal and multiprogramming environment. To achieve the multi-terminal support required by ORS, a FORTRAN callable asynchronous terminal driver was developed.

DJ-11 TERMINAL DRIVER

The DJ-11 driver provides ORS with the following features:

1. Support of 16 terminals concurrently.
2. Overlapped I/O on all terminals.
3. Automatic type-ahead facility with bell-ringing to signal input echo buffer full.
4. Selection of a logical line to connect to a physical line and disconnection of a logical line from a physical line (i.e. dynamic reconfiguration of lines).
5. A "no-echo" facility on input (particularly useful for password security).
6. Field delete capability.
7. Attention key capability.
8. Overlapped program execution with all terminal I/O.

With this driver, event-driven structured programming in FORTRAN is possible. In addition, it provides a means of controlling data transferred to specific terminal ports thus allowing total crt screen formatting control.

An initial call to the driver is required prior to any other subroutine calls. This is done through a routine called DJINIT. Its main function is to set interrupt vectors and initialize the DJ-11 control registers. Its counterpart is DJTERM — a termination routine to be called just before exiting to the RT-11 system. This, in effect, disables all ports on the multiplexor.

Before any I/O can be done on a terminal, a connection must be made between the actual physical port number and the user defined logical number, thus allowing the user complete and dynamic control over the terminal's configuration. DJENAB and DJDSAB are the subroutines used to connect and disconnect terminals. Figure 3 shows a possible port configuration for the ORS system. In this example, two hardcopy terminals are assigned to physical lines 00 and 02 on the multiplexor. In turn, they are logically connected to these lines via the line numbers 01 and 02. Similarly, the crt terminals are logically 03 and 04 and physically 09 and 12. All I/O to these terminals are done by using only the logical numbers assigned to them.

When an I/O is required to a specific terminal, a call is made to the driver, passing it an event flag variable, the logical line number assigned to the terminal and the buffer variable into or from which the I/O is to occur. The operation is initiated by the driver (providing the previous I/O has been completed and the requested terminal does exist) and control is returned immediately to the calling program. The following is an example of a calling sequence to read information from a terminal, echoing each character as it is entered.

```
CALL  DJREAD (IFLAG, LOGLNE, BUFF, CNT)
```

IFLAG is an integer variable used as the event flag for determining the status of the current read operation for the requested terminal. LOGLNE is an integer variable containing the logical number of the terminal. The variable BUFF will be used to store the characters entered and CNT, an integer variable, will contain the maximum number of characters to be entered prior to the call to DJREAD and the actual number of characters read once a carriage return is entered at the terminal. Prior to the call, IFLAG is set to zero to indicate that the I/O is in progress. Execution continues in the calling program from the point at which the call was made. Once the I/O is complete, the driver will set IFLAG for that terminal to indicate the completion. It is up to the user to poll IFLAG and wait for the I/O completion before initiating another I/O on that same terminal.

The subroutine DJRDNE is the same in calling sequence and in use as DJREAD with the exception that the characters entered are not echoed at the terminal. This routine is particularly useful for implementing password

114

LOGICAL VS PHYSICAL TERMINAL ASSIGNMENTS

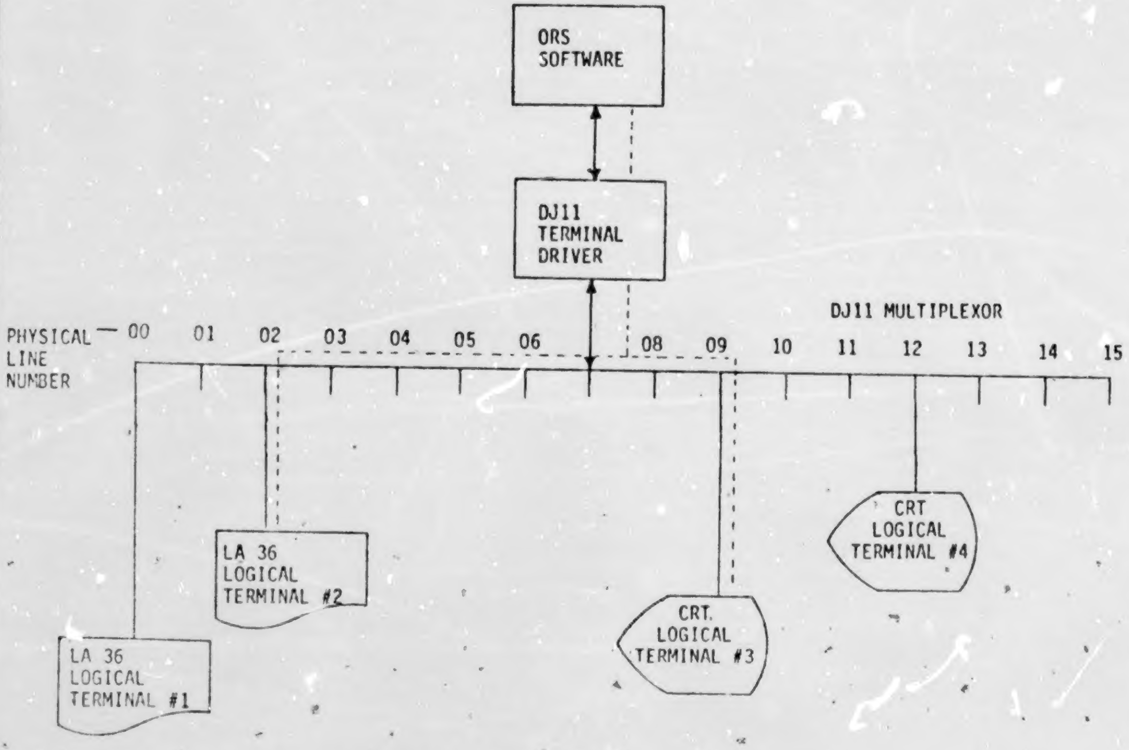

ORS SOFTWARE
DJ11 TERMINAL DRIVER
DJ11 MULTIPLEXOR
PHYSICAL LINE NUMBER
00
01
02
03
04
05
06
08
09
10
11
12
13
14
15
LA 36 LOGICAL TERMINAL #2
LA 36 LOGICAL TERMINAL #1
CRT. LOGICAL TERMINAL #3
CRT LOGICAL TERMINAL #4

Figure #3 Sample port configuration used in ORS.
Note: Dashed line (---) represents a logical connection between two terminals.

security. ORS uses this feature for exactly that function.

To send information to a terminal, the subroutine DJWRTE is used. It has a calling sequence similar to DJREAD.

```
CALL DJWRTE (IFLAG,LOGLNE,BUFF)
```

The variables IFLAG and LOGLNE serve the same functions as they did in the DJREAD calling sequence. In this routine, though, BUFF contains the ASCII character string that is to be sent to the terminal instead of received. The calling program controls all carriage returns and line feeds to allow the user to format a crt screen in whatever manner desired. To keep consistent with the string manipulation functions in RT-11, DJWRTE requires that the character string be terminated with a null character. Since output continues until the null character is encountered, multiple lines can be displayed with one call of DJWRTE by merely inserting carriage return-line feed characters in the buffer whenever needed.

Figure 4 shows a very simple use of the DJREAD and DJWRTE calls. Assuming that the line has been successfully enabled, the event flag for logical terminal number 2 is set to zero to indicate an I/O operation is in progress. The first operation encountered is the call to the driver to read a maximum of 30 characters from the terminal. Once the driver has initiated the request, control is returned to the calling program and the statements after the call statement are executed. Since it is desired to send back to the terminal the entire line entered, it is necessary to wait for the read operation to complete. The following statement, taken from the program shown in Figure 4, accomplishes this function.

```
10     IF(IFLAG.NE.1) GO TO 10
```

At first glance, this statement appears to loop forever. Since the completion routine DJREAD is concurrently executing and upon completion will set IFLAG to 1, execution past this loop will eventually occur. Then, and only then, can another I/O operation, namely DJWRTE, be initiated.

EVENT-DRIVEN FORTRAN

Figure 4 shows the transfer of data to and from a single terminal. The code could be extended to include multiple terminals, but the structure shown will not support them effectively. Essentially, the major fault with the structure is that as the system is polling IFLAG for completion on the DJREAD request for one terminal, the other terminals would be idle, waiting for servicing. The wait state would last until a carriage return character was received from the first terminal being serviced (which could be an infinite amount of time). A very simple solution that is used in ORS, is to structure the code to facilitate round-robin dispatching. Figure 5 shows the main dispatcher for ORS. Loop 35, starting with statement 0040 and ending with statement 0090, is an infinite loop which can only be terminated through a special command entered at the console terminal. The DO loop, starting with statement 0041 and ending with statement 0089, allows polling of all enabled terminals. It should be noted that the variable J, found

```
      LOGICAL*1 BUFF(31)
          .
          .
          .
      IFLAG=0                              !SET EVENT FLAG TO SHOW I/O IN PROGRESS
      LOGLNE=2                             !LOGICAL LINE NUMBER ASSIGNED TO TERMINAL
      CNT=30                               !SET MAXIMUM CHARACTER COUNT
      CALL DJREAD (FLAG,LOGLNE,BUFF,CNT)   !INITIATE A READ ON TERMINAL
          .
          .
          .
10    IF (IFLAG.NE.1) GO TO 10             !POLL EVENT FLAG FOR COMPLETION
      BUFF(CNT+1)=0                        !TERMINATE INPUT CHARACTER STRING WITH NULL
      IFLAG=0                              !RESET EVENT FLAG FOR ANOTHER I/O
      CALL DJWRTE (IFLAG,LOGLNE,BUFF)      !SEND CHARACTER STRING TO TERMINAL
          .
          .
          .
20    IF (IFLAG.NE.1) GO TO 20             !POLL EVENT FLAG FOR COMPLETION
          .
          .
          .
      STOP
      END
```

Figure #4 Sample code showing the use of subroutines DJREAD and DJWRTE to a single terminal.

```
35    CONTINUE
      DO 1000 J=1,I
      TTFLG=80    !SET CONSOLE CHAR LIMIT
      CALL CKCON(TTFLG,TTBUFF)     !POLL CONSOLE TERMINAL
      IF(TTFLG.EQ.0)GO TO 40       !BRANCH IF NO INTERRUPT FROM CONSOLE
      CALL TTOUT(I)                !ENTER SUB. IF INTERRUPT HAS OCCURRED
40    IF(IFLG(J).EQ.0)GO TO 1000   !BRANCH IF I/O STILL PENDING
      IF(IFLG(J).NE.101) GO TO 200 !BRANCH IF NO CNTL/C HIT
C...
C...  IF CNTL/C WAS HIT, CONTROL WILL BE TRANSFERRED TO "ATTN"
C...  UNLESS TERMINAL WAS LOGGED OFF.  THEN CONTROL WILL RETURN
C...  TO "LOGON".
C...
      IF(IP(J).EQ.1.AND.REG1(J).EQ.4)GO TO 190
      IF SVIP(J).NE.0)GO TO 180
      SVIP(J)=IP(J)                !SAVE PROGRAM POINTER
180   IP(J)=7                      !CHANGE PROG. POINTER FOR "ATTN"
      REG1(J)=1            !RESET SUBROUTINE ENTRY POINTER
      GO TO 200
190   REG1(J)=5
      CALL ICMKT(J,TMPB)           !CANCEL TIMER
      ISW(J)=0                     !RESET SWITCH FOR NO TIMER
200   CONTINUE
C...
C...  BRANCH TO REQUESTED PROGRAM
C...
      GO TO (100,300,400,500,550,600,700),IP(J)
100   CALL LOGON(J)
      IF(REG1(J).NE.4)GO TO 100    !BRANCH IF NOT READY FOR TIME
      N(J)=1                               !SET CHAR LIMIT
C...
C...  THIS READ WILL NOT ECHO ANY CHARACTERS TO THE TERMINAL &
C...  WILL MAINTAIN THE TERMINAL IN THE LOGGED OFF STATE UNTIL
C...  A CONTROL/C IS HIT.
C...
      CALL DJRDNE(IFLG(J),ITERM(J),BUFF(1,J),N(J))
      IF(IPR(J).NE.3)GO TO 1000    !BRANCH IF TABLE POINTER NOT 3
      IF(ISW(J).NE.0)GO TO 1000    !BRANCH IF TIMER SWITCH HAS BEEN SET
C...
C...  START TIMER.  "TMEOUT" WILL BE CALLED WHEN 1 MINUTE IS UP.
C...
      CALL ITIMER(0,1,0,0,AREA,J,TMEOUT)
      SVRG(J)=REG1(J)              !SAVE SUBROUTINE ENTRY POINTER
      SVIPR(J)=IPR(J)              !SAVE TABLE POINTER
      ISW(J)=1                 !SET TIMER SWITCH TO INDICATE ACTIVE
      GO TO 1000
300   CALL REGALL(J)
      GO TO 1000
400   CALL CKSTAT(J)
      GO TO 1000
500   CALL CALLST(J)
      GO TO 1000
550   CALL FEES(J)
      GO TO 1000
600   CALL UPDCAL(J)
      GO TO 1000
700   CALL ATTN(J)
1000  CONTINUE
      GO TO 35
      STOP
      END
```

Figure #5   Round Robin Dispatcher for ORS.

throughout the ORS software, represents a pointer to the terminal currently being serviced. With these two loops, all terminals can be continuously polled in a round-robin fashion for service requests. The actual dispatching is done within the loops and the choice of which routine is to be entered is determined by the current program state of the terminal contained in variable IP. Everytime the DO loop in statement 0041 is entered, the console terminal flag is checked for pending service requests from other terminals. Statement 0047 is very important to the concept of round-robin dispatching. All terminals which are considered to be in the idle state are actually idle in the read mode either through a DJREAD or DJRDNE call to the driver. Therefore, the event flag for that terminal is always zero (I/O in progress) until a request needs to be serviced. Statement 0047 allows the dispatcher to check the next terminal in the polling sequence for any pending requests, when the terminal it is currently checking is awaiting I/O completion. This is also true when a DJWRTE has been activated by the I/O has not been completed. Again, the event flag will be zero. Statement 0063 controls the actual dispatching of a task according to the current program state held in variable IP. Once control is returned to the dispatcher, the next terminal in polling sequence will be checked for possible servicing. When a service request is dispatched, control is passed to a subroutine with the terminal pointer as a parameter. This routine may itself be a dispatcher and in turn pass control to another routine. Although control is initially passed into the same location every time the subroutine is called, the same section of code may or not may be executed. In essence, the code within the routine is broken down into modules. Each module performs specific tasks which constitute an event once control returns to the main dispatcher. The order in which these modules are executed is maintained by the pointer REG1 specifically for terminal J. This modularity provides ORS with the pseudo-reentrant code needed for a multiprogramming environment and a method of providing "time-slicing" to the main dispatcher without using specific time quantums. Note that ORS is one executable program in which entry and reentry into task-oriented modules make it appear as a multiprogramming system.

An example showing pseudo-reentrant code is found in Figure 6. It is a section of code from the LOGON subroutine used in ORS. Statement 0011 determines which section of code is to be executed for the current service request. It is controlled by the event pointer contained in REG1. When called for the first time, execution in LOGON will start at statement labeled 25, second time at 50, and so on. After each module has been executed, the event pointer is updated to reflect which event will occur next when the subroutine is reentered. Control is then passed back to the main dispatcher.

AN OVERVIEW OF ORS

As shown in Figure 7, ORS consists of a main dispatcher, a console terminal handler, a DJ-11 multiplexor handler, several application subroutines and a variety of shared service routines. When the system is started, all the terminal buffers, search tables, control blocks and program modules are loaded into memory (see Figure 8). The initial configuration of the

```
      SUBROUTINE LOGON(J)
      LOGICAL*1 USERID(5,16),PASWRD(5,16),PSWST(16),BUFF(83,6),
     XPGRPAS(5,16),PGWST(16),IPRG(6),FF,CR,LF,ZERO,BLK,BKSP,
     XALPHS
      INTEGER IP(6),IPR(6),ORDER(40,50),N(6),RLEN(40,5),
     XPROC(16),REG1(6),IFLG(6),ITERM(6),USCTR,PSCTR,IUSER(6)
      COMMON/IDCTR/USERID,PASWRD,PSWST,PGRPAS,PGWST,USCTR,PSCTR,PROC
      COMMON/PRCTR/ORDER/BUFFER/BUFF/LGCTR/N,RLEN
      COMMON/DJCTR/IFLG,ITERM/FLCTR/IPR,IP,IPRG
      COMMON/CHCTR/BLK,CR,LF,ZERO
      COMMON REG1,ICN,IUSER
      DATA FF,BKSP/"014,"010/,ALPHS/'S'/
      JM=1
      GO TO (25,50,75,510,100,200,250),REG1(J)
C...
C... CLEARS SCREEN WITH FORM FEED
C...
25    BUFF(1,J)=FF
      BUFF(2,J)=0
      CALL DJWRTE(IFLG(J),ITERM(J),BUFF(1,J))
      GO TO 500
C...
C... WRITES HEADER PROMPT TO TERMINAL
C...
50    CALL DJOUT(J)
      IPR(J)=IPR(J)+1
      GO TO 500
C...
C... WRITES DATE AND SECOND HEADER PROMPT
C...
75    DO 74 K=1,83
74    BUFF(K,J)=BLK                 !CLEAR BUFFER
      BUFF(1,J)=CR                  !INSERT CARRIAGE RETURN
      BUFF(2,J)=LF                  !INSERT LINE FEED
      CALL DATE(BUFF(3,J))          !GET CURRENT DATE FROM SYSTEM
      ICN=ORDER(IPR(J),IP(J))       !SET RELATIVE RECORD POSITION FOR PROMPT
      READ(16'ICN)(BUFF(K,J),K=23,59)  !GET PROMPT
      CALL TIME(BUFF(71,J))         !GET CURRENT TIME FROM SYSTEM
      DO 78 K=76,80
78    BUFF(K,J)=BKSP                !INSERT BACKSPACES
      BUFF(81,J)=0                  !TERMINATE BUFFER WITH NULL
      CALL DJWRTE(IFLG(J),ITERM(J),BUFF(1,J))
      IPR(J)=IPR(J)+1
      GO TO 500
          .
          .
          .
500   REG1(J)=REG1(J)+1
510   RETURN
      END
```

Figure #6   Example of Pseudo-reentrant Code

OVERVIEW OF ORS

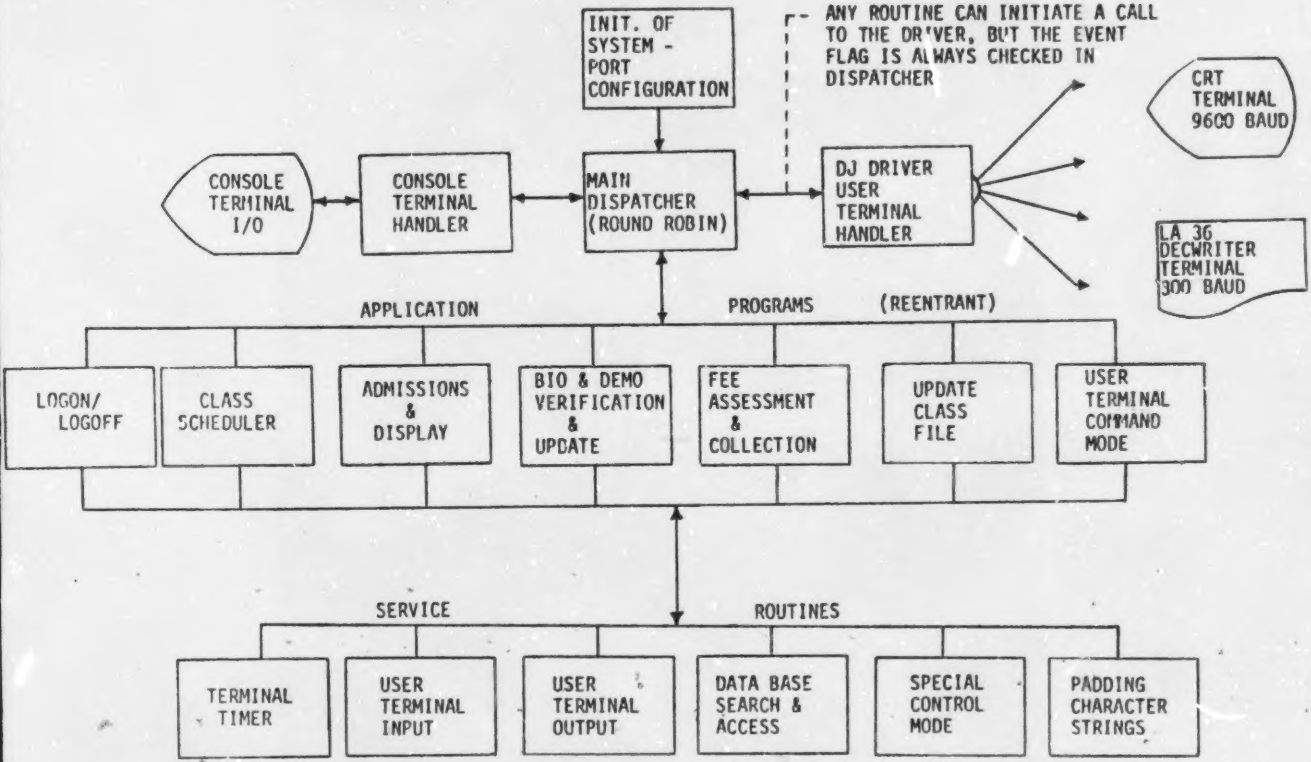
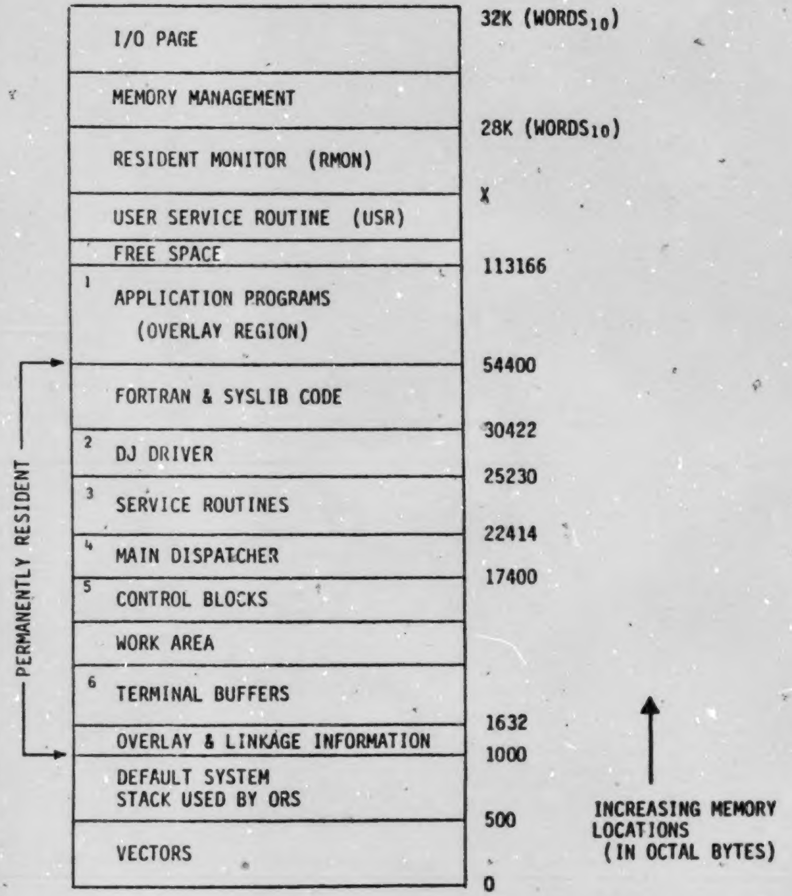

INIT. OF SYSTEM - PORT CONFIGURATION
ANY ROUTINE CAN INITIATE A CALL TO THE DRIVER, BUT THE EVENT FLAG IS ALWAYS CHECKED IN DISPATCHER
CRT TERMINAL 9600 BAUD
CONSOLE TERMINAL I/O
CONSOLE TERMINAL HANDLER
MAIN DISPATCHER (ROUND ROBIN)
DJ DRIVER USER TERMINAL HANDLER
LA 36 DECWRITER TERMINAL 300 BAUD
APPLICATION
PROGRAMS
(REENTRANT)
LOGON/ LOGOFF
CLASS SCHEDULER
ADMISSIONS & DISPLAY
BIO & DEMO VERIFICATION & UPDATE
FEE ASSESSMENT & COLLECTION
UPDATE CLASS FILE
USER TERMINAL COMMAND MODE
SERVICE
ROUTINES
TERMINAL TIMER
USER TERMINAL INPUT
USER TERMINAL OUTPUT
DATA BASE SEARCH & ACCESS
SPECIAL CONTROL MODE
PADDING CHARACTER STRINGS

Figure #7 Software modules and their configuration within the ORS System.

BACKGROUND ENVIRONMENT FOR O.R.S.


I/O PAGE
32K (WORDS$_{10}$)
MEMORY MANAGEMENT
28K (WORDS$_{10}$)
RESIDENT MONITOR (RMON)
X
USER SERVICE ROUTINE (USR)
FREE SPACE
113166
1
APPLICATION PROGRAMS (OVERLAY REGION)
54400
FORTRAN & SYSLIB CODE
30422
2
DJ DRIVER
25230
3
SERVICE ROUTINES
22414
4
MAIN DISPATCHER
17400
5
CONTROL BLOCKS
WORK AREA
6
TERMINAL BUFFERS
1632
OVERLAY & LINKAGE INFORMATION
1000
DEFAULT SYSTEM STACK USED BY ORS
500
VECTORS
0
PERMANENTLY RESIDENT
INCREASING MEMORY LOCATIONS (IN OCTAL BYTES)

1 BUSINESS & REGISTRATION AUDIT TRAILS
CLASS SCHEDULER
BIO & DEMO DATA ENTRY (ADMISSIONS)
BIO & DEMO VERIFICATION & UPDATE
FEE ASSESSMENT & COLLECTION
LA 36 OUTPUT
COMMAND MODE FOR USER TERMINALS
TWO-WAY MESSAGE SYSTEM
LOGON/LOGOFF PROCEDURES
CONSOLE TERMINAL CONTROL
CLASS FILE UPDATE

2 ENABLE & DISABLE LINES
OVERLAPPED I/O ON ALL TERMINALS
SPECIAL FUNCTION KEYS

3 MAINTAINS TERMINAL I/O
ALLOWS FOR SPECIAL CONTROL MODE
SERVICES MAIN CONSOLE TERMINAL

4 ROUND-ROBIN SCHEDULER
WITHOUT TIME SLICING

5 TABLES CONTROLLING:
SCREEN FORMATTING
PROMPT OUTPUT
BUFFER POINTERS
REENTRANT POINTERS

6 STUDENT MASTER
CLASS DATA
STUDENT OVERFLOW AREAS
TERMINAL I/O (USER & CONSOLE)

Figure #8 Shows detailed locations of software used once ORS is located users' terminals is entered through the ORS system console and all requested terminals are enabled.

Once the system is "up" all terminals are in the "logged off" state with the date and current time (which is updated every minute) displayed for the convenience of the user. The log-on procedure can be initiated with the attention key after which the prompt

USERID:

is displayed. To maintain protection against illegal entry into the system, the user must first give ORS his user identification code, then his password and finally the program's password in which he will be working. In addition, his password type must match the type given to the terminal. Once in a program, the user will be lead through that program with an ordered set of prompts. As an example, the admissions procedure first prompts the user with the following question:

DO YOU WISH TO DISPLAY?

The user will then respond with a yes or a no. With either response, ORS will ask the user for the student's identification number. The system will first see if the requested student's record is currently in use. This record lockout maintains data integrity for all data structures found in the system. If the record is currently in use, the user is warned with a message at his terminal and the program is immediately restarted. Otherwise, ORS will try to locate the student's record. If found and a display was requested, all the student's biographic and demographic information is displayed at the terminal. If the student's record was not found and a display was not requested, this information can be entered through the terminal item per item as ORS prompts the user for each item.

At anytime the user can get the attention of the system by hitting the attention key. This puts the user's terminal into command mode. The following is a list of commands currently available to the user.

HE - Displays all commands
LO - Logoff current user
RE - Restart current program
RS - Resume current program
NP - Return to password for a new program
B' - Broadcast message to console terminal
SD - Display current date
ST - Display current time

Note the the "NP" command gives the user complete control over which program is to be entered.

The following is a list of programs found in ORS. Each entry is accomplished by a brief description of its function.

Logon/Logoff - a security program which allows a user with the proper identification to use the ORS.
Class Scheduler - a program to display, modify or create class

schedules for students admitted to PCC. Builtin safeguards prevent duplicate course scheduling and illegal add/drop modifications. Credit hour assessment and file status are set.

Admissions - a program to either enter or display a student's biographic and demographic information.

Admissions Verification and Update - allows the user to verify and update a student's biographic and demographic information either on an item per item basis or according to a group of items.

Fee Assessment and Collection - a program to clear either by cash, check or through accounts receivable any outstanding fees incurred by registered students. Prints out receipt along with the student's current schedule.

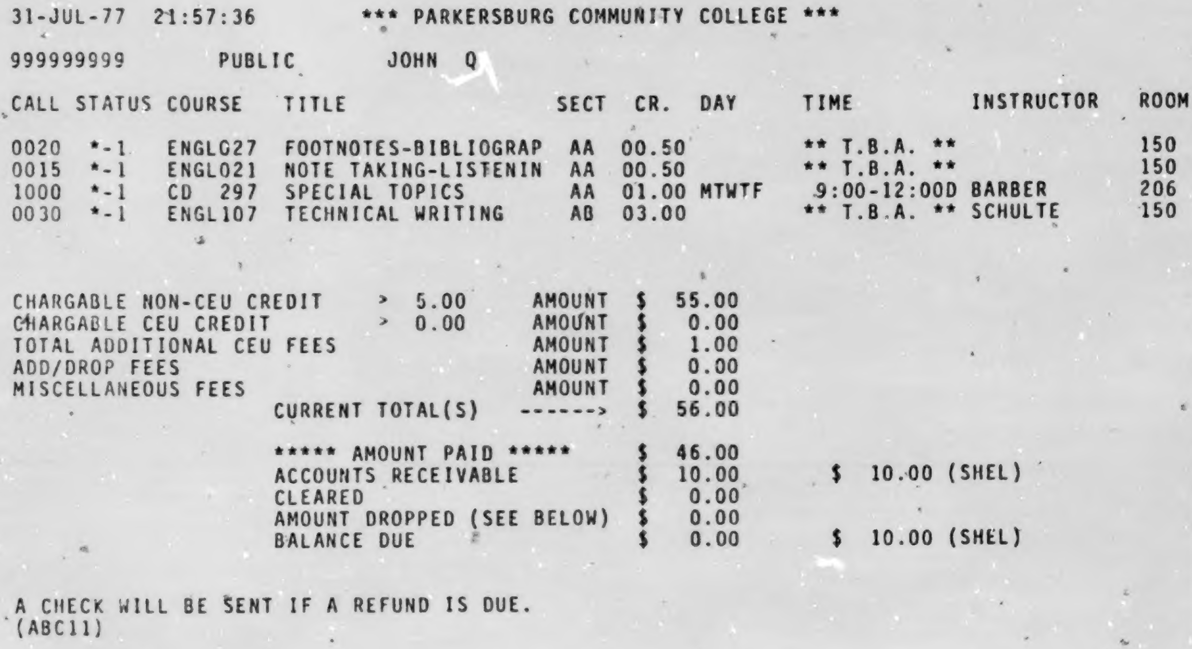
Class File Update - allows the entering and updating of course information.

User Terminal Command Mode - allows the user to direct program flow, send messages to host terminal and display current date and time.

The new student who registers under the ORS system must follow a prescribed set of procedures. First, he must be admitted to the college. Upon acceptance, a master record containing all his biographic and demographic information is created. Second, a course schedule must be drawn up and approved. It is then entered into the system. Third, the student must clear all business transactions created by the registration. Once cleared, the student receives a schedule and receipt similar to the one shown in Figure 9. In this example, John Q. Public signed up for four courses. The call numbers, course names and descriptions, sections, credit hours, times, instructors and room assignments are listed as a convenience to the student (Note: T.B.A. means to be announced). The actual charge assessed against the student is listed along with his payment arrangements.

CONCLUSIONS

The need for on-line computing services is increasing very rapidly today. Hardware costs are on the decline and the trend is toward smaller main frames with large amounts of memory — certainly favorable for on-line applications. But many facilities are still hampered with limited computing power. Event driven FORTRAN provides a means for simulating the multi-terminal and multiprogramming environments on these machines. It is fast, efficient and rather conservative in memory use. The concept, itself, is a very powerful one and has lead to the creation of the ORS system. The versatility and success of this system has proven that batch-oriented systems still in use today are not the only way to provide academic services. The possibilities are limitless.



```
31-JUL-77  21:57:36        *** PARKERSBURG COMMUNITY COLLEGE ***

999999999       PUBLIC       JOHN  Q

CALL STATUS COURSE   TITLE                  SECT  CR.  DAY      TIME          INSTRUCTOR   ROOM

0020   *-1   ENGL027  FOOTNOTES-BIBLIOGRAP   AA  00.50         ** T.B.A. **                 150
0015   *-1   ENGL021  NOTE TAKING-LISTENIN   AA  00.50         ** T.B.A. **                 150
1000   *-1   CD  297  SPECIAL TOPICS         AA  01.00 MTWTF    9:00-12:00D BARBER          206
0030   *-1   ENGL107  TECHNICAL WRITING      AB  03.00         ** T.B.A. ** SCHULTE         150


CHARGABLE NON-CEU CREDIT     >  5.00      AMOUNT  $  55.00
CHARGABLE CEU CREDIT         >  0.00      AMOUNT  $   0.00
TOTAL ADDITIONAL CEU FEES                 AMOUNT  $   1.00
ADD/DROP FEES                             AMOUNT  $   0.00
MISCELLANEOUS FEES                        AMOUNT  $   0.00
                    CURRENT TOTAL(S)     ------>  $  56.00

                    ***** AMOUNT PAID *****       $  46.00
                    ACCOUNTS RECEIVABLE           $  10.00      $  10.00 (SHEL)
                    CLEARED                       $   0.00
                    AMOUNT DROPPED (SEE BELOW)    $   0.00
                    BALANCE DUE                   $   0.00      $  10.00 (SHEL)

A CHECK WILL BE SENT IF A REFUND IS DUE.
(ABC11)
```

Figure #9   Sample output from ORS.

SNAP/SHOT YOUR ABILITY TO SUPPORT THAT NEXT APPLICATION

Ernest L. Jones, Director
Data Systems and Services
Indiana University

SNAP/SHOT (System Network Analysis Program-Simulated Host Overview Technique) is a discreet simulation of a network and/or host model available through IBM at the Raleigh System Center. The simulator provides an analysis of a total IBM Communications System including a maximum of 100 CPU's and their associated 370X, remote 370X, lines and terminals.

Input data must be obtained from RMF, SMF and the CICS Analyzer to determine the existing environment and from analysis and users on anticipated applications and their volumes. The simulator permits the introduction of numerous variables in order to evaluate the impact of altering resources such as memory, software, paging devices, CPU's, lines and terminals and altering volumes of transactions, message mixes, size of transactions, etc. The results provide response times by transaction, by line; shows CPU and line utilization, software and 3705 time. Bottlenecks are easy to spot, and the impact of changes to eliminate the bottlenecks can be evaluated by changing parameter and getting another run.

Indiana University has used SNAP/SHOT twice to identify what volume of transactions was supportable on existing equipment and what changes would improve performance. After the first run, a group was assigned to optimize some transactions; and additional memory was added to reduce the paging rate. The second run was made to see what was accomplished and to simulate simultaneous student registration at the six small campuses and then finally adding the large campuses.

Graphs and charts contained in the paper identify the many variables utilized in the processing of Indiana University data and show the results obtained.

If you have an uneasy feeling about the impact of some new, large project about to be placed on your computer--whether it is a new on-line transaction, RJE, lines and terminals, etc.--you might want to consider "SNAP/SHOTing" it. SNAP/SHOT (Systems Network Analysis Program-Simulated Host Overview Technique) is a discreet simulation of an IBM Communications network and/or host model. The simulator provides an analysis of a maximum of 100 CPU's and their associated 370X, remote 370X, paging devices, lines, terminals, communications protocol, and associated software.

To use SNAP/SHOT, the user provides a description of each application to be included in simple, macro terms such as the number of instructions executed, the number of DB/DC calls, the number of I/O's, line configurations, types and numbers of terminals, line speeds, communication controllers, modems, and paging devices. More details and sources of information will be discussed later. The first step is to establish a base level in the simulator results that compares favorably with observations of actual performance; therefore, you specify the central processor in use and the paging rate you are observing on your system at some selected time period. One item that is not specified, which seems strange at first, is memory size. After some thought, it is clear that paging rate is directly related to the amount of memory available and the philosophy of the installation on the use of memory.

After all input data has been provided and SNAP/SHOT has been executed, the host analysis section provides information on CPU utilization based on path lengths, interrupts, priorities, and the CPU type specified. The network section using the Emulator and/or Network Control Program line scheduling algorithm calculates line utilizations, message response times, and queues. This model can generate a diagram of the network described and provide the rate of messages generated for each location.

Overall message rates for the system are user specified as a series of parameters prior to each run, thus allowing future or peak loads to be simulated.

Indiana University has utilized SNAP/SHOT twice. The first time was in March, 1977, and the second time in September, 1977. We had discussed an earlier use, but could not supply information in sufficient detail to drive the model.

The sequence of events that led to the first use began when several on-line systems were implemented in quick succession in the summer of 1976. Each of these systems when tested on the test data bases with test volumes seemed to operate within acceptable parameters, but when thrown together with production volumes was completely unacceptable. Tuning efforts were continuous during the fall and winter months, but response time on the student records maintenance transactions remained at 18 seconds average with wide fluctuations.

Some equipment and software changes had been identified which we thought would improve response time, but we could only speculate on the degree of improvement that would result. We had identified that the paging rate during the time the on-line systems were operational was higher than it should be. A crash program had been proposed to review all major transactions for the purpose of optimizing their performance, but again we were not sure how much we would have to improve these programs to make significant improvement. A goal of eight second response time was set on the complex Student Records Maintenance and Admissions transactions. Retrievals were to be in the two to four second response-time range.

Armed with our input for the model, our IBM Systems Engineer headed for Raleigh. A number of sources had been tapped to accumulate the data, including software packages, system programmers, application analysts and users. As mentioned above, a crash program had been started to identify response time problems, so a large part of the technical fact gathering was in place or completed.

Since users are an important part of any computer application, let's look at their participation in the projection of the work loads.

The registrar provided a list of each regional campus, the number of students to be enrolled, the number of days and hours per day of registration. With these figures the number of students to be processed per hour and the required number of terminals were calculated.

The application analyst worked with the programmers to determine the number of CICS messages and tasks to perform a complete student registration. Combined with the above statistics a current message rate, as well as a growth factor, resulted in a message rate for each campus by line and terminal. The 3705 Emulation Program (EP) Generation provided each existing line, speed, number and types of terminals, and type of modem used.

The SMF listings for each RJE station recorded the number of cards read, lines (characters) printed and jobs run by given time periods. Several days were processed and the peak periods were used for the simulation. Also the batch requirements were obtained from the SMF output.

CICS Performance Analyzer (PA) was used for many needed facts about the transactions and terminals. A report by terminal gave us the current transaction or application mix and the number of characters in and out. This concluded the data necessary to define our network to the simulator.

PA provided most of the needed information about the applications. A listing by application gave us the CPU seconds, from which we could calculate the approximate number of instructions executed, the number of I/O operations and the number of tasks completed. The total number of tasks was used to approximate the total CICS load by application.

Together with the system programmers we completed the CPU and Systems definitions.

From all of the above sources the needed input for the simulator was then encoded in the proper macros.

Before we left for Raleigh a meeting with system programming, application systems and the SE refined the data to our best estimate for both the current system and the projected changes.

The primary goal of the March visit to Raleigh was to pinpoint what was causing our response time problems and to identify where our efforts should be concentrated to help alleviate the problems.

Once the data was input and we understood the procedures necessary to execute the program, we set out to firmly establish the base run that would represent--as closely as possible--our observation of the actual system.

Fortunately the Raleigh Center provided a consultant, who was one of the authors of the system, to answer questions as they arose. On Monday and Tuesday we called him frequently on technical as well as philosophical questions. By Wednesday morning we felt we had established a base level. We later confirmed this by calling the Assistant Director of System Programming and Operations and running a controlled experiment for a four-hour period where we eliminated a large portion of our batch initiators and attempted to control our paging rate at 15 pages a second.

Prior to describing some of the capabilities of SNAP/SHOT that we used, it is necessary to describe the form that was designed to record the data for comparison purposes. Please see the example and explanation in Appendix A.

Forty runs were made in three days and they are described in Appendix B. A review of that list of runs will indicate that we compared hardware, operating systems, memory size (paging rate), line speeds and protocol, and paging devices.

The effect of holding all other factors constant while reducing the paging rate (increasing available memory) is shown in Graph 1. This action alone will not "save the ship" but will give the crew some time to patch the holes. This is exactly what was done. A software team was established to review all transactions and make changes to improve their performance. The number of keyed elements on the data base was reduced from 69 to 22; and every program was run through OPTIMIZER II, a product of CAPEX. The team, affectionately known as the Software Attack Team or SWAT, accomplished more than had been hoped. This will be described later.

Graphs two, three, and four represent simulations of different central processors at 10 and 20 pages a second. Obviously the Attached Processor provided considerable improvement in performance over the Unit Processor in both on-line response time and batch throughput, and at ten pages a second outperformed the 158MP.

The 168-1 and 168-3 obviously would be the machines to have to improve response time without requiring a lot of programmer time. However, funds were not available; and we were not on IBM's delivery schedule.

Although some runs were made, little effort was spent reviewing line speeds and organizations because it was obvious we were unable to sustain a message rate high enough to tax the existing line configuration. One area of the output of SNAP/SHOT identifies the point of major delay as shown in Chart 1. If you read from right to left beginning with the column headed "APPL PROG," you can see the time delay at each point. This particular example identifies lines as the major problem.

The second trip, made in September, 1977, had two goals: 1) Evaluate the improvements made through SWAT and hardware additions (1 meg. of memory) and 2) to determine if the on-line registration system could be supported on all campuses when the AP was installed. The feeling was that the largest campus with 32,000 students was out of the question. The Indianapolis campus with 26,000 students was doubtful, but the small campuses could be handled.

Since we had become "experts" during our first trip to Raleigh, we were scheduled for Monday-Wednesday on the second trip. We had ample time to accomplish our goals. Twenty-one runs were completed which are described in Appendix C.

The first seven runs were made on various mainframes after establishing the base run on the 158-3. Hardware simulated was 158-3AP, 168-1, 168-3, 168-3AP, and 3033. The operating system we used in all cases was MVS, and the paging rate was held to 10 pages per second. These decisions were firm after our first trip. Increased message rates were used to reflect a growing environment.

One of our runs during our first trip used a 15% reduction in the number of instructions executed in each transaction which indicated that response time could be improved considerably if the transactions were optimized, and the Software Attack Team set about to do just that.

Graph 5 represents simulations before and after SWAT. We held all factors constant except the description of the program logic which we changed based on the improved versions of the programs as a result of our optimizing efforts. Work was done on both on-line transactions and batch programs. The on-line results represent the improvement made in the Student Records Maintenance transactions. Again the results were confirmed through observations made in actual operational situations.

Working from the new base, the changes in central processors produced the results in Graph 6. The 158AP deteriorates very quickly as the message rate increases. Both on-line and batch capability are impacted. The 3033 maintains the best response time of all systems, but the 168AP provides additional batch capability. From a cost/

performance ratio, the 3033 is still the best buy. If we see some 168AP machines appear in the third-party market, the cost/performance gap may close.

At this point we added the Registration transactions and increased the message rate to include what we anticipated the volume would be if all six small, remote campuses were to register simultaneously. These campuses range from 700 to 6300 students. We limited our simulations to the 158AP which we planned to install in December, 1977, and the 3033 which was the machine size chosen to take us into the future.

We estimated that we would experience a message rate of 1.0 - 1.1 per second, and the results indicated that the 158AP should handle this message rate without significant degradation. The 3033 would show no impact on response time at that level.

The Indianapolis campus was then added to the transaction mix which we felt would generate 1.3 - 1.5 messages per second. It is obvious from Graph 7 that the 158AP would produce undesirable results. If the message rate did reach 1.5 per second, erratic conditions would exist with a 45 second response time and a standard deviation of 8.59. The 3033, on the other hand, remains relatively constant.

Adding the Bloomington campus to the "registration" mix produced some interesting results. At a message rate of 1.7, the response time was almost 8 seconds greater than at 1.7 messages per second with the small campuses plus Indianapolis. Because the registration transactions are larger in terms of resources required than the transactions normally used at registration by the Bloomington campus, communication lines became the constraining factor. A quick look at factors presented by SNAP/SHOT led us to experiment with changing from BTAM to NCP/VTAM and evaluating the effects of the new 32XX terminals using SDLC.

The ability to multidrop cathode ray tubes and RJE stations led to a line configuration which indicated we could go from 22 lines to six 9600 baud lines and drop our response time on the full registration implementation to 2.70 seconds at 1.35 messages per second.

Where are we now? This past semester with two small campuses on the registration system, we experienced erratic response times on the 158AP. What happened? Prior to September, 1977, our average message rate per second for a period of a month was .46 to .50--obviously with peaks and valleys--so we forecast a doubling of that rate for registration. An analysis of the three days of registration this fall uncovered a significantly higher message rate, reaching 2.18 messages per second in one 15-minute period but averaging during the three days better than 1.5 per second. New transactions have been added, and the Bloomington campus' use of on-line transactions during registration was significantly higher than anticipated.

We are hoping to get another SNAP/SHOT this winter in order to apply our new information on transaction volumes and try to forecast what is going to happen when we add a new Purchasing Inquiry system, a Student Loan system, Financial Aids system, Course Maintenance, etc.

The synergistic effect could be devastating. Transactions tested on Test CICS and approved, can, when put with all other transactions on Production CICS, cause serious problems. We plan to push the 3033, lines, paging devices, etc. to their limit to see what growth potential we have with this machine, which will be installed in July, 1979.

In summary, SNAP/SHOT provides an opportunity to look into the future at what impact new applications or growth of existing ones will have on your ability to service your users and where you might look to make changes in hardware or software that would allow you to cope with the new environment.

NOTES ON READING OUTPUT REPORTS

The following is a description of the comparison charts which are used in the report:

Columns	Explanation
1	Run number - each simulation was given a unique number which can then be used as a quick reference. The run numbers and description of each follows the charts.
2	Paging Rate - the paging rate of the CPU for a given simulation.
3	MSGS/SEC - the actual number of messages per second for the given simulation.
4	RESPONSE TIME - RM - the average total response time for an "RM" transaction. Total time is the sum of application time, 3705 time and line time for "RM" transaction only.
5	Std Dev - the value of one standard deviation from the average total response time of an "RM" transaction.
6	AD - the average total response time for and "AD" transaction.
7	Std Dev - the value of one standard deviation from the average total response time of an "AD" transaction.
8	CPU Utiliz - the average CPU utilization for the duration for this particular simulation. (In the case of an Attached Processor System there are two CPU's; and thus, two utilization percentages.)
9	BATCH COMPL - based on the criteria of the advanced job, this column is total of batch jobs completed in all batch regions during the simulation.
10	CICS & DB Util - the sum of utilization for the CICS/application region and the data base region.
11	RM APL TIME - the time it took the "RM" application to execute in the CPU.

Appendix A

Columns	Explanation
12	AD APL TIME - the time it took the "AD" application to execute in the CPU.
13	LINE 34 RM - the total response time of an "RM" transaction on line 34 only.
14	LINE 34 AD - the total response time of an "AD" transaction on line 34 only.

PERFORMANCE OF 158-3 UP SVS WITH VARYING RATES

Run Number	Paging Rate	Msgs./ Sec.	Response Time				CPU Utiliz.	Batch Compl.	CICS & DB Util	RM Apl. Time	AD Apl. Time	Line 34 RM AD
			RM	StdDev	AD	StdDev						
1	20	.41	12.9	3.4	15.0	5.0	93.4	50.5				
		.46	18.0	11.0	26.0	14.6	94.4	44.7				
		.51	24.0	14.7	28.7	16.4	95.6	37.2				
		.67	57.9	22.3	58.7	22.8	97.9	20.2				
38	15	.41	11.5	2.9	13.8	5.1	93.0	51.7				
		.46	14.2	8.5	21.9	11.9	94.2	47.7				
		.51	18.6	10.1	22.9	10.1	95.1	38.6				
		.67	42.3	21.1	43.4	21.7	98.2	17.2				
4	10	.41	10.5	2.4	12.5	4.7	92.8	52.4				
		.46	12.6	7.7	17.9	9.1	93.5	50.2				
		.51	17.2	9.7	20.2	9.1	94.6	40.0				
		.67	34.4	18.2	36.7	18.6	98.5	13.9				

RUN NO.	CPU TYPE	OPERATING SYSTEM	PAGING RATE	DESCRIPTION
1	158-3 UP	SVS	20	Base (=message mix and terminal configuration of existing environment).
	Purpose:	Establish performance criteria for current system.		
2	158-3 UP	MVS	20	Base.
	Purpose:	Determine impact on current system of changing to MVS operating system without adding additional memory.		
3	158-3/AP	MVS	20	Base.
	Purpose:	Establish performance criteria for current loads on an AP system with a paging rate of 20.		
4	158-3 UP	SVS	10	Base.
	Purpose:	Same as #1 with reduced paging rate (indicative of additional memory).		
5	158-3 UP	MVS	10	Base.
	Purpose:	Same as #2 with lower paging rate.		
6	158-3/AP	MVS	10	Base.
	Purpose:	Same as #3 with lower paging rate.		
7	158-3/AP	MVS	10	Increased message rates on base.
	Purpose:	Determine performance of AP at higher utilization.		
8	Not Used	(See Note 1)		
9	Not Used	(See Note 1)		
10	158-3 UP	SVS	10	Addition of VTAM/NCP and SDLC on base.
	Purpose:	To determine impact of an SNA environment on current system.		
11	158-3 UP	SVS	10	Elimination of all batch work.
	Purpose:	Evaluate impact of changing current system to all TP environment.		

Appendix B

RUN NO.	CPU TYPE	OPERATING SYSTEM	PAGING RATE	DESCRIPTION
12	158-3/AP	MVS	10	Increased message rates on base.
	Purpose:	Add additional points for graphing purposes to Runs 6 and 7.		
13	158-3/AP	MVS	20	Use drum for a paging device.
	Purpose:	Determine significance of a drum in a high paging environment.		
14	168-3	MVS	20	Base.
	Purpose:	Determine performance of workload on a 370/168 Model 3.		
15	168-1	MVS	20	Base.
	Purpose:	Same as #14 with a 370/168 Model 1.		
16	158-3/AP	MVS	20	Base.
	Purpose:	Evaluate performance of an AP with no batch running (see #11).		
17	158-3/AP	MVS	20	Increased message rates.
	Purpose:	Same as #7 with a higher paging rate.		
18	158-3/MP	MVS	10	Base.
	Purpose:	Evaluate MP performance.		
19	158-3/AP	MVS	10	Change to a DL1/CICS environment with HDAM files.
	Purpose:	Experiment with a DL1 interface with CICS using HDAM file construction in the data base.		
20	158-3/AP	MVS	20	Base with 15% fewer instructions.
	Purpose:	Simulate effect of a 15% greater efficiency in application programs.		
21	158-3AP	MVS	30	Base with 15% fewer instructions.
	Purpose:	Same as #20 with higher paging rate.		

RUN NO.	CPU TYPE	OPERATING SYSTEM	PAGING RATE	DESCRIPTION
22	158-3 UP	SVS	20	No batch.
	Purpose:	Same as #11 with higher paging rate.		
23	158-3 UP	SVS	10	Only 3 batch initiators.
	Purpose:	Determine effect on current system of cutting back to only 3 initiators at a low paging rate.		
24	Not used (see note 2)			
25	168-1	SVS	20	Base.
	Purpose:	Same as #15 with SVS operating system.		
26	158-3/AP	MVS	10	Increased message rates on base.
	Purpose:	Same as #7		
27	158-3/AP	MVS	10	Maintaining current message rates, line 34 was changed to 1 control unit with 24 terminals running at 4800 baud, SDLC.
	Purpose:	Determine impact of SDLC in environment described.		
28	158-3/AP	MVS	10	Same as #27 at 9600 baud, SDLC.
	Purpose:	Determine impact of higher baud rate with SDLC.		
29	158-3/AP	MVS	10	Same as #27 at 4800 baud, bisync.
	Purpose:	Determine performance of environment described in Run #27 with existing 4800 baud, bisync facilities.		
30	158-3 UP	SVS	10	Change to a DL1/CICS environment with HDAM files.
	Purpose:	Experiment with a DL1 interface with CICS using HDAM file construction in the data base, running on current equipment.		
31	158-3 UP	SVS	10	Same as #30, but using HIDAM files.
	Purpose:	Same as #30, but with HIDAM file construction in data base.		
32	158-3/AP	MVS	10	Same as #27 at 9600 baud, bisync.
	Purpose:	Determine impact of increasing to 9600 baud, bisync in environment described in #27.		

RUN NO.	CPU TYPE	OPERATING SYTEM	PAGING RATE	DESCRIPTION
33	158-3/AP	MVS	20	Using a 3350 for a paging device.
	Purpose:	Determine significance of using a 3350 in a high paging environment.		
34	158-3AP	MVS	10	New mix of transactions with additional terminals added (total of 158 terminals).
	Purpose:	Determine performance of AP in future anticipated environment.		
35	158-3/AP	MVS	10	Message rates of approx. 50% increased volume on line 34 (1 control unit, 26 terminals) run at 4800 baud, bisync.
	Purpose:	Determine performance of a 4800 baud, bisync line in environment described.		
36	158-3/AP	MVS	10	Change to a full IMS DB/DC environment.
	Purpose:	Experiment with AP performance in a full IMS environment.		
37	158-3/AP	MVS	10	Same as #35 with a 9600 baud, bisync line.
	Purpose:	Same as #35 at 9600 baud, bisync.		
38	158-3 UP	SVS	15	Base.
	Purpose:	Attempt to duplicate exact situation being experienced on current system on 3/17/77.		
39	158-3/AP	MVS	10	New mix of transactions, additional terminals, local lines.
	Purpose:	Determine significance of putting some terminals on local lines.		
40	158-3/AP	MVS	10	Same as #35 with a 4800 baud, SDLC line.
	Purpose:	Same as #35 with a 4800 baud, SDLC line.		

Note 1: Initial attempts to determine significance of 9600 baud lines, but not properly configured.

Note 2: An initial run to determine significance of a 3350, but run at an improper paging rate.

RUN NO.	CPU TYPE	MESSAGE MIX	DESCRIPTION
1	158-3	C	Base (message mix, load, and terminal configuration of existing environment).
	Purpose:		Establish performance criteria of current system.
2	158-3AP	C	Base.
	Purpose:		Establish performance criteria for current loads with the AP installed.
3	158-3AP	C	Base with increased message rates of current mix.
	Purpose:		Determine if there is a breaking point on the AP with current mix.
4	3033	C	Base.
	Purpose:		To provide a comparison of the 158 and the 3033, also to establish a base for the 3033.
5	168-1	C	Base.
	Purpose:		To establish if the 168-1 should be in the migration plan.
6	168-3	C	Base.
	Purpose:		To establish if the 168-3 should be in the migration plan.
7	168-3AP	C	Base.
	Purpose:		To establish if the 168-3AP should be in the migration plan.
8	158-3AP	CR	Base with registration mix for regional campuses.
	Purpose:		To determine the load on the CPU with regional campus registration and the respective line utilizations.
9	3033	CR	Base with registration mix for regional campuses.
	Purpose:		Same as #8 on a 3033

RUN NO.	CPU TYPE	MESSAGE MIX	DESCRIPTION
10	158-3AP	CRI	Base with registration mix for regional campuses and IUPUI.
	Purpose:		Same as #8 with addition of IUPUI.
11	3033	CRI	Base with registration mix for regional campuses and IUPUI.
	Purpose:		Same as #9 with the addition of IUPUI.
12	Not Used		
13	3033	T	Base with registration mix for all campuses including Bloomington.
	Purpose:		To determine the load on the CPU and lines for full registration.
14	158-3AP	T	Base with full registration and NCP/VTAM, SDLC and new terminals.
	Purpose:		To determine impact of new terminals under SDLC on the 158-3AP.
15	3033	T	Base with full registration mix, BTAM and old terminals.
	Purpose:		Same as #14 on a 3033.
16	Not Used		
17	3033	T	Base with full registration mix, VTAM, new terminals, and increased batch.
	Purpose:		To determine if additional batch loads could be handled and the impact on TP.
18	3033	T	Base with full registration and NCP/VTAM, SDLC and new terminals and new line configurations at 4800 baud.
	Purpose:		To establish a base of multidrop lines and their impact on response time.

RUN NO.	CPU TYPE	MESSAGE MIX	DESCRIPTION
19	3033	T	Same as #18
	Purpose: Same as #18 but with a simulation of 30 minutes.		
20	Not Used		
21	3033	T	Same as #18 with 9600 baud.
	Purpose: To determine the benefits of increased line speed.		

Message Mix Explanation:

C = Current message mix.

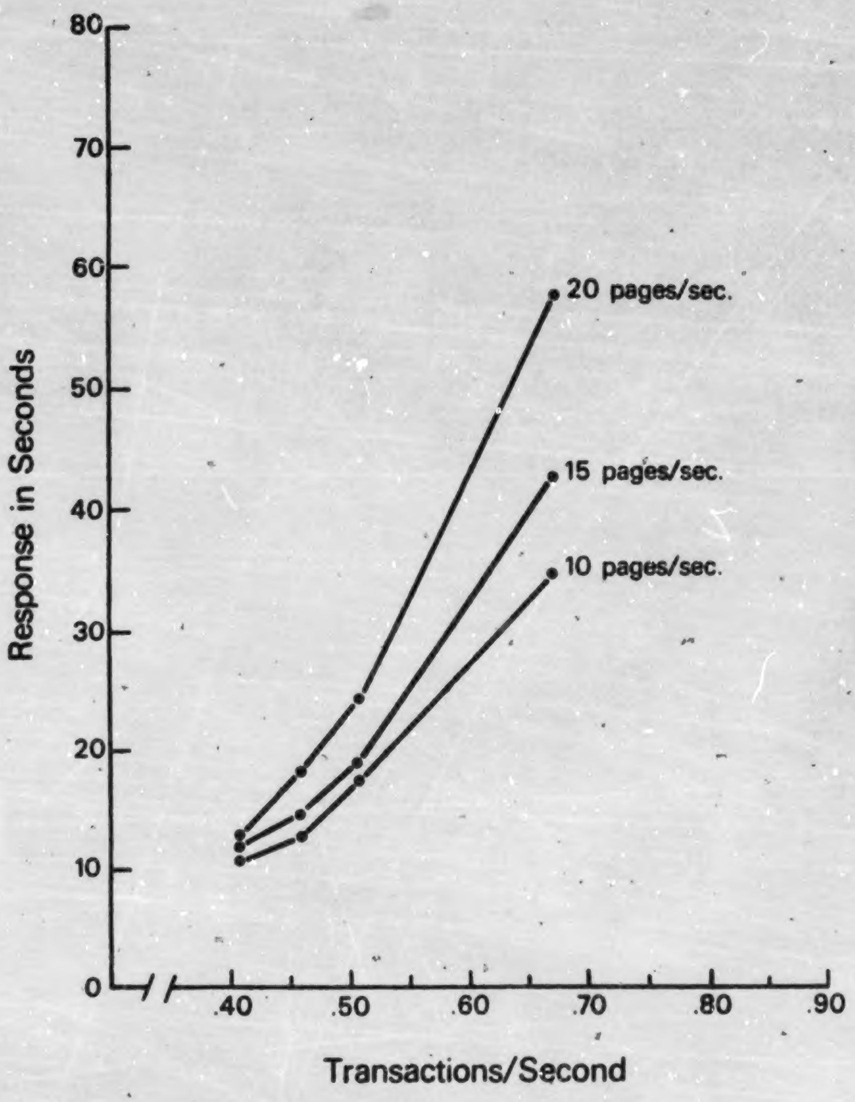
CR = Current message mix with predicated regional campus registration load.

CRI = Current message mix with predicted regional campus and IUPUI registration load.

T = Current message mix with full predicted registration load at all campuses including Bloomington.

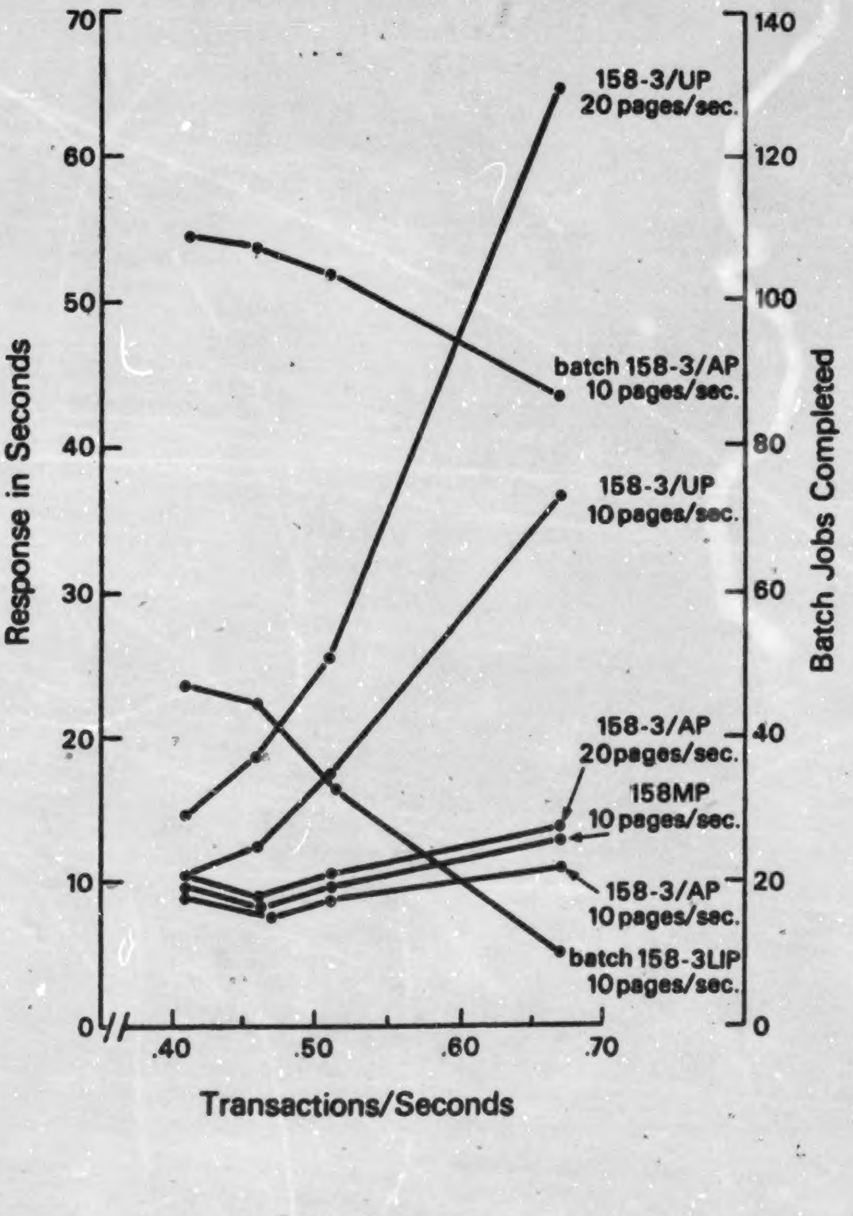
Effect of Lower Paging Rate (Increased Memory)
Records Maintenance Transaction

158-3 Unit Processor
SVS Operating System



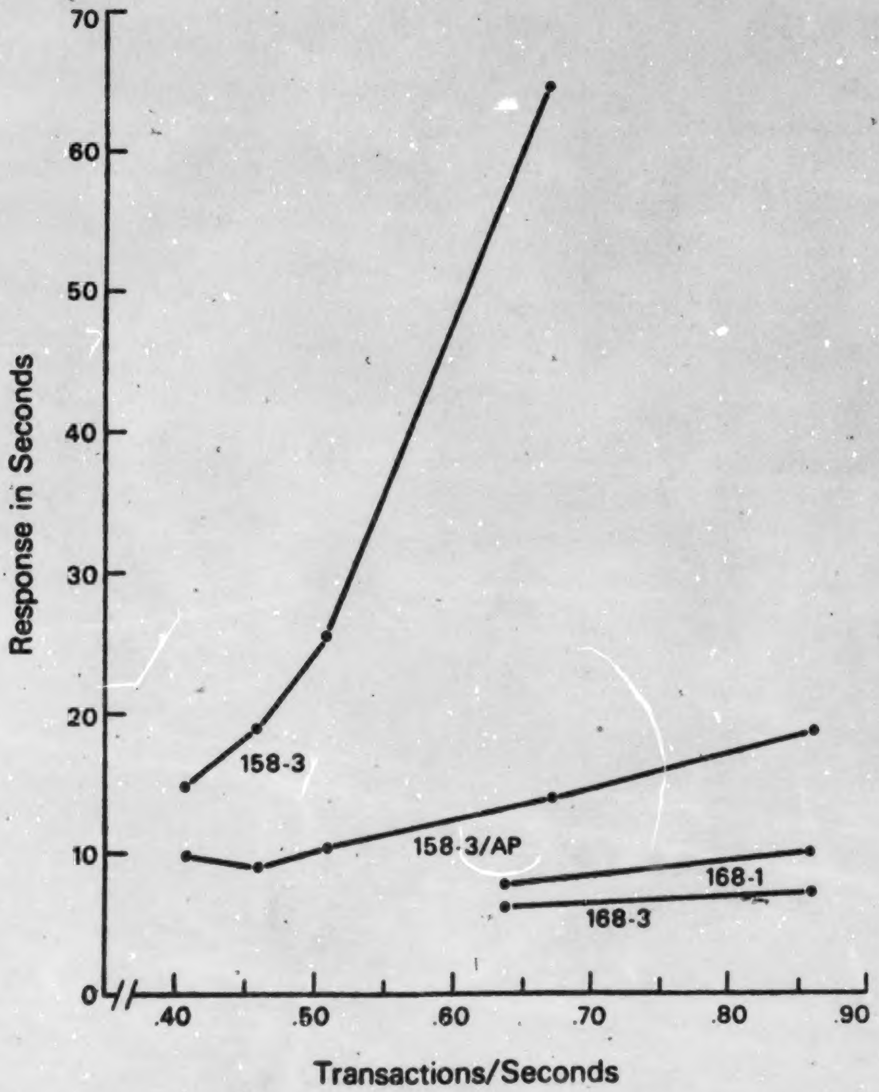
Graph 1

158-3/UP Compared to 158-3/AP
MVS Operating System



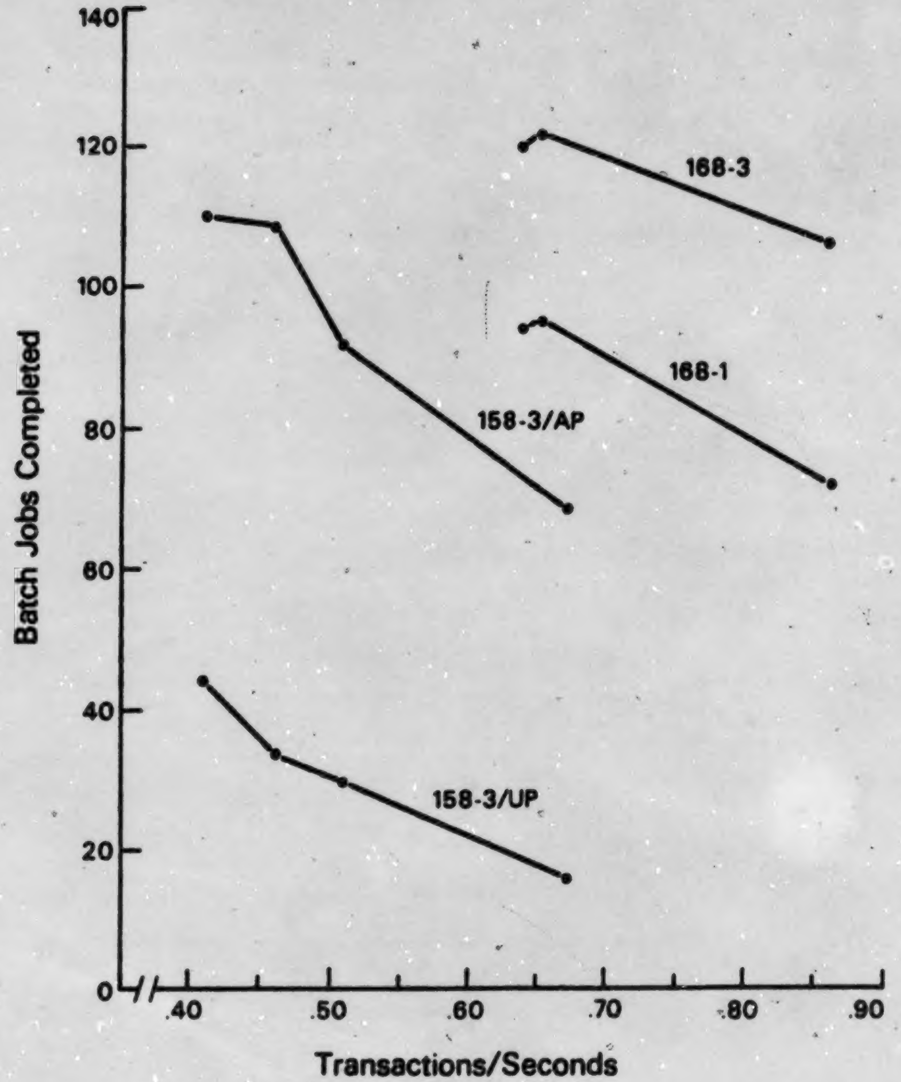
Graph 2

**Effects of Different CPU's
Records Maintenance Transaction**

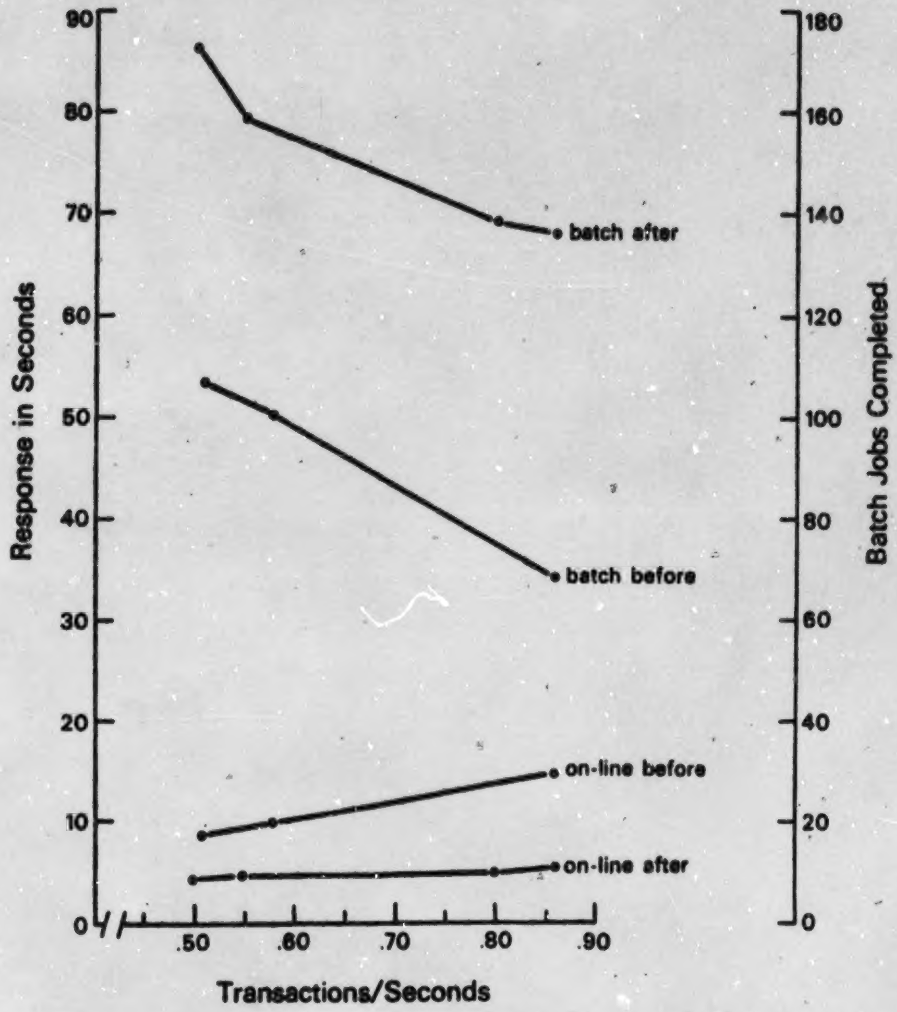


Graph 3

Effects of Different CPU's
Batch Through-Put

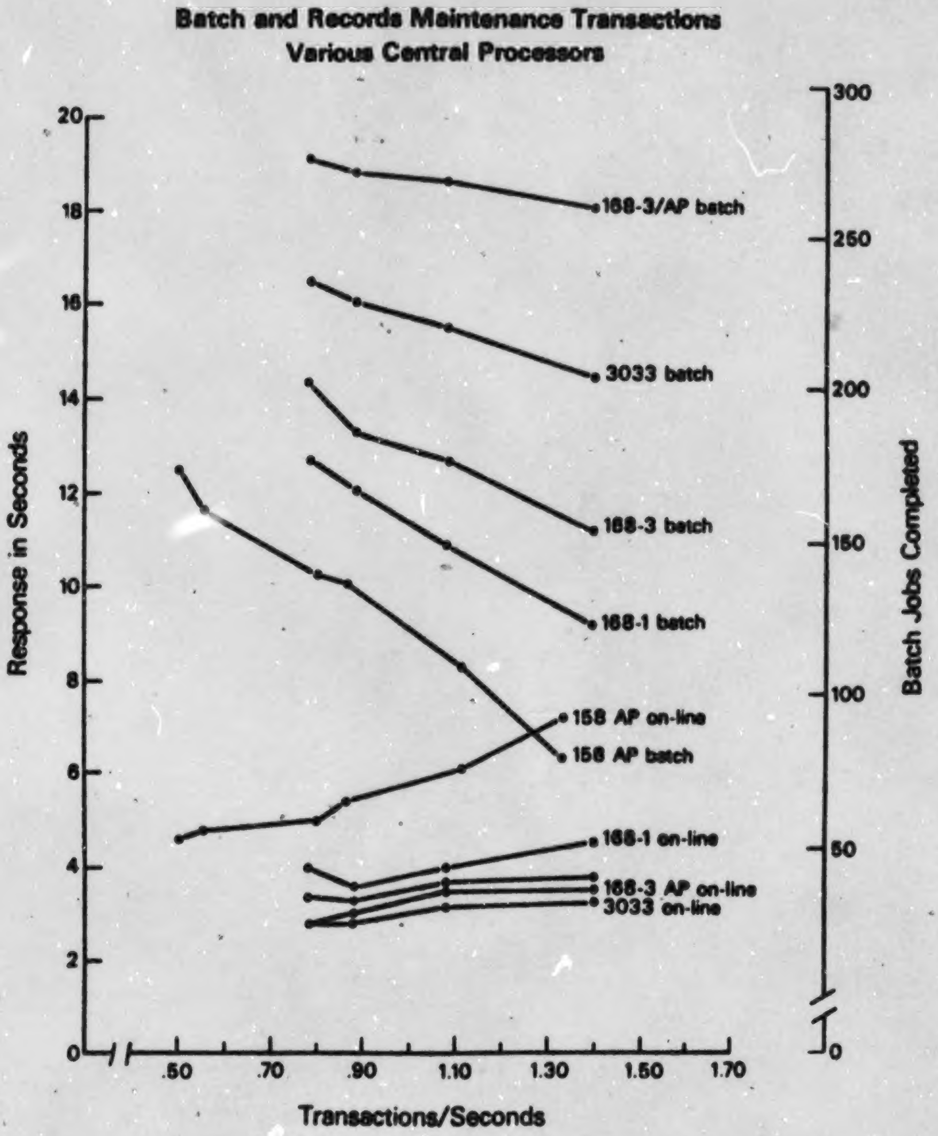


Graph 4



Before and After SWAT
158-3/AP 10 pages/sec.
MVS Operating System

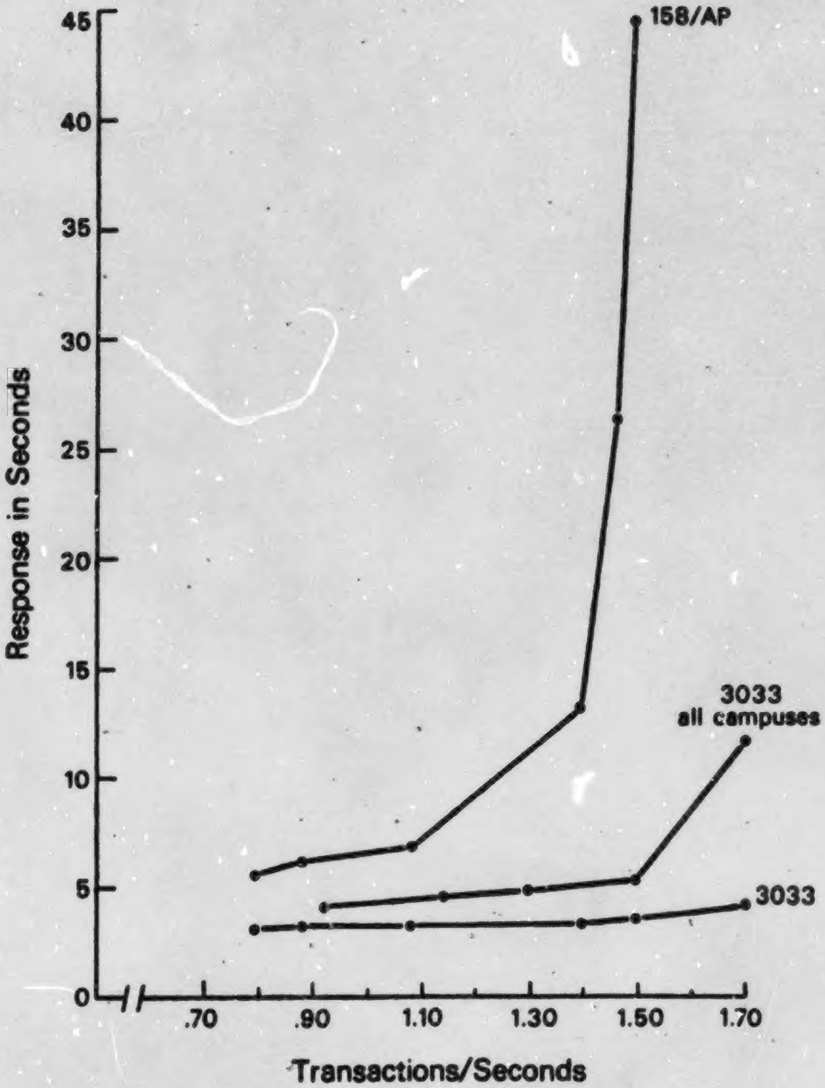
Graph 5


Batch and Records Maintenance Transactions
Various Central Processors
Response in Seconds
Batch Jobs Completed
Transactions/Seconds
0
2
4
6
8
10
12
14
16
18
20
.50
.70
.90
1.10
1.30
1.50
1.70
0
50
100
150
200
250
300
168-3/AP batch
3033 batch
168-3 batch
168-1 batch
158 AP on-line
158 AP batch
168-1 on-line
168-3 AP on-line
3033 on-line

Graph 6

Registration at the Small Campuses and Indianapolis

158/AP vs 3033 and 3033 All Campuses



Graph 7

RESPONSE TIMES FOR EACH NAMED MESSAGE IN SECONDS.

MESSAGE.NAME	TOTAL SAMPLE	LINE	3705	3705 NET-WORKING	CPU	CPU NET-WORKING	APPL PROG
ADMSG	45	7.012	4.069	3.820	3.818	3.814	3.814
CLMSG	29	5.245	3.870	3.504	3.503	3.499	3.499
RMMSG	33	5.774	3.962	3.709	3.709	3.706	3.706
SCMSG	3	6.143	4.177	3.774	3.774	3.770	3.770
SSMSG	17	6.683	3.919	3.854	3.854	3.851	3.851

Chart 1

MODELING AND SIMULATION WITH INS

Stephen D. Roberts, Ph.D.
Director
Associate Professor

James Schier
Programmer/Analyst

Thom Sulanke
Programmer/Analyst

Health Systems Research Section
Regenstrief Institute/Department of Medicine
Indiana University School of Medicine
Indianapolis, Indiana, 46202

ABSTRACT

Networks can be used to describe a large class of systems including health care, manufacturing, and computer systems. INS, the Integrated Network Simulation language, provides a set of symbols (nodes and branches) corresponding to familiar processes such as queues and activities which can be combined to create a network model of a system. The resulting network provides an excellent means of communicating the structure of the system to other people. The modeler translates the INS network into INS statements which are then read and executed by INS. INS automatically provides comprehensive statistical output on the behavior of the model. Based on the output, the original model can easily be altered and extended. Furthermore, experiments can be conducted in which characteristics of the network are changed. The result of these experiments can then be used to make predictions about the effects of making similar changes to the system being studied. INS is powerful yet easy to use and teach and requires no prior knowledge of simulation or programming. A textbook on INS is available. INS is written in standard FORTRAN for portability.

INTRODUCTION

Computer simulation is increasingly being used to aid decision-makers in a variety of environments including manufacturing, production, service, and public sectors. Building a simulation model can be greatly simplified by using a simulation language. For many languages, the simulation modeler must place the problem into a next event structure, code events, collect statistics, monitor status, and format reports. Hence the modeler must be a programmer who is also well founded in simulation concepts. In addition, the resultant simulation model will have limited meaning to anyone other than the model builder.

INS, the Integrated Network Simulation language, overcomes many of these deficiencies by putting simulation modeling into a network framework and automatically performing the programming activities mentioned above. The INS modeler conceives the simulation problem graphically as a network, composed of predefined modeling symbols (nodes and branches). These modeling symbols correspond to familiar processes such as queues and activities and they enable the modeler to easily abstract directly from the real system to the INS network. The symbols are translated to INS statements. INS automatically executes the simulation and provides output. Consequently, the INS modeler can focus on the problems of modeling and not be impeded by programming tasks. INS requires no competence in computer programming.

Since INS is concise, easy to document, and visually appealing, INS models are also excellent vehicles for communication between modelers and their clients. Such communication encourages the decision-makers to actively participate in the modeling process, thus increasing the probability of eventual implementation.

The purpose of this paper is to illustrate the use of INS in model building and systems analysis. Space limits us to a discussion of a single example and a few embellishments. You are referred to the textbook (1) that fully describes the language and includes examples to illustrate its many modeling features.

BASIC CONCEPTS IN INS

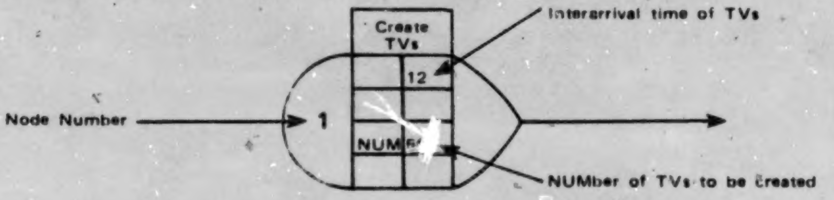
The INS network is constructed about the flow of units of traffic called transactions. A transaction is a generic term which is interpreted by the modeler. For example, transactions may represent TVs coming into an inspection station, people arriving for hair cuts, ships entering a harbor, or customers coming into a gas station. The nodes and branches within the network depict creating transactions, routing them through the network, queuing at activities, servicing at activities, and eventually departing from the network.

Transactions may require resources to process them in the network. The resource in INS is the generic term applied to an entity which responds to the demands of transactions within the network. In INS the resource can also exercise independent decision-making in fulfilling its service requirement. Examples of resources are inspectors who inspect TVs, barbers who cut hair, tugs which assist ships into a harbor, and gas station attendants and pumps which service customers.

A TV INSPECTION AND ADJUSTMENT PROBLEM

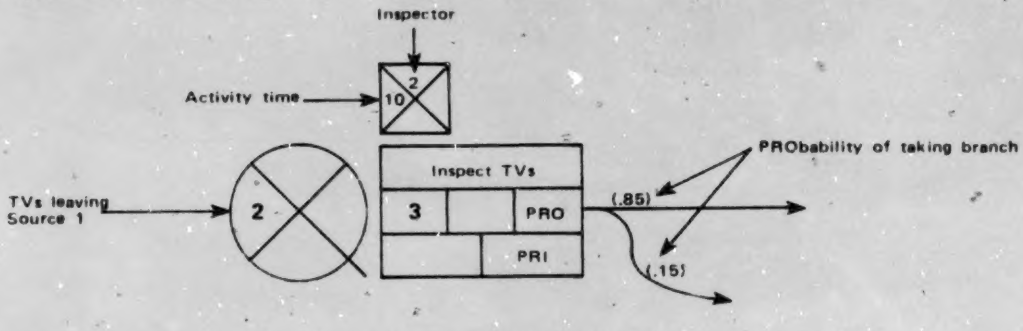
As a portion of their production process, TV sets are sent to a final inspection station at the rate of one set every 12 minutes. The inspector spends 10 minutes examining a TV. Of the sets inspected, 85% pass but 15% fail and are sent to an adjusting station. The adjustor repairs the set in 30 minutes and sends it back for reinspection where the set again has a 15% chance of failure.

This problem is easily interpreted as a network. We begin by identifying the entities involved. Transactions will be used to represent TVs since they are the units of traffic demanding service in the system. The inspector and adjustor are resources since they provide service to the TVs and constrain the rate at which TVs are processed. The TVs will enter the network at a node called a source node. The source node controls when and how many TVs will arrive. Let us say we want 500 TVs to be created at 12 minute intervals. The source node looks like the following:


Interarrival time of TVs
Create TVs
12
Node Number
1
NUM
NUMber of TVs to be created

Every node is given an identifying node number appearing on the left hand side of the node. The arrow pointing out of the node is called a branch and indicates the node to which the TVs must go next. In this manner, nodes are connected to form the network.

Inspection and adjustment of TVs are represented by activity nodes. Activities are places where transactions (TVs) are delayed, usually to receive service. The inspector and adjustor constrain the processing and no more than one TV at a time can be inspected or adjusted. Thus, other TVs are forced to wait for service in queue nodes. Queue and activity nodes are always adjacent to each other so that no branching is required. For the inspection activity, the queue and activity nodes look like this:


Inspector
Activity time
10
2
PRObability of taking branch
Inspect TVs
TVs leaving Source 1
2
3
PRO
PRI
(.85)
(.15)

TVs wait in Queue 2 (shaped like a large Q) until the inspector (resource number 2) can inspect them. Following the 10 minute inspection at Activity 3, 85% of the TVs leave via one branch while 15% leave via the other branch heading for a similar queue and activity representing TV adjustment.

TVs which successfully pass inspection are no longer needed and they leave the network at a sink node:

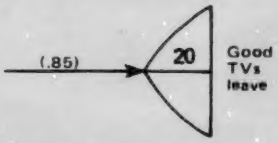
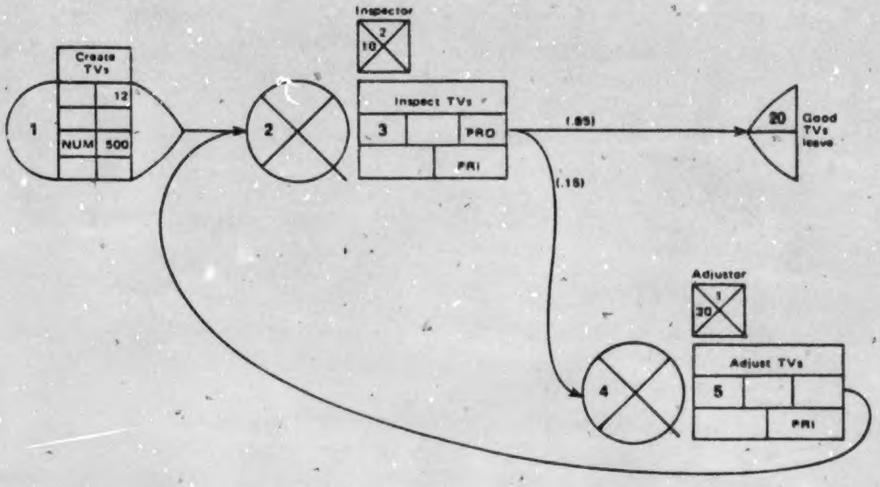

(.85)
20
Good TVs leave

Figure 1 shows the nodes assembled to form the entire network.

Figure 1: Basic TV Inspection and Adjustment Problem


Create TVs
1
12
NUM 500
2
Inspector
2
10
Inspect TVs
3
PRO
PRI
(.85)
(.15)
20
Good TVs leave
Adjustor
1
30
4
Adjust TVs
5
PRI

RESOURCES AVAILABLE	
Description	Number
Adjustor	1
Inspector	2

Figure 2: INS Statements for the Basic Network

```
RESOURCE,1,1                               $ Adjustor
RESOURCE,2,2                               $ Inspector
$
SOURCE,1,,,12,,,NUMBER OF TVS =,500        $ Create TVs
    BRANCH TO, 2
$
QUEUE,2                                    $ Wait for inspector
ACTIVITY,3,PRO,,,,,PRI                     $ Inspect TVs
    GROUP,1,2,,,10                         $ Using the inspector for 10 minutes
    BRANCH TO,15, .85                      $ 85% pass
    BRANCH TO, 4, .15                      $ 15% fail
$
QUEUE,4                                    $ Wait for adjustor
ACTIVITY,5,,,,,,PRI                        $ Adjust TVs
    GROUP,1 ,1,,,30                        $ Using the adjustor for 30 minutes
    BRANCH TO,2                            $ Return for re-inspection
$
SINK,20                                    $ Good TVs leave
```

We have used the nodes as building blocks, arranging them to agree with our conception of how the real system works. We have connected the nodes with branches to characterize the flow of transactions from node to node thereby allowing us to create models of any size or complexity out of simple, basic processes. Changes in individual processes are confined to single nodes. Furthermore, entire models can be easily altered by adding, subtracting, or rearranging the nodes and branches. The network is intuitively appealing since it corresponds visually to our understanding of the real system.

Before we demonstrate the ease of enhancing the basic model, let us see how our network is translated into a form suitable for the computer. Each node, branch, and resource in Figure 1 is translated to an INS statement by the direct correspondence of specifications on the node to fields on a statement. Fields in INS statements are separated by commas. Thus, the network in Figure 1 becomes the set of statements in Figure 2. The dollar sign ($) is used to indicate that no more fields exist, allowing comments to be added which help to document and clarify the network. The use of this feature makes the INS input very readable. The third field of the SOURCE and ACTIVITY statements is used to control the method of branching. TVs branch from Activity 3 by PRObability; at other nodes, TVs go directly to the indicated node. Other methods of branching (not shown) allow transactions to test system conditions such as queue length and current time. The GROUP statement corresponds to the square symbol above the activity node and tells INS which resources are capable of servicing transactions at the activity as well as the activity time. Multiple commas are used to skip unused fields corresponding to the blank specifications on the node.

Expanding the TV Inspection and Adjustment Problem

Let us now embellish our system and see how it affects the model. Suppose the inspection of TVs is divided into two parts. The first part requires an inspector to use a circuit testing machine while the second part requires only the inspector. Also assume a second, more experienced inspector is now available. This inspector can also adjust TVs but prefers to inspect them. When given a choice between inspecting a newly arrived TV and one which has been adjusted, this older inspector prefers to reinspect the adjusted TV. If both of the inspectors are available when a TV arrives to be inspected, the newer inspector should be selected leaving the more highly skilled inspector available for other tasks. Finally, the activity times for inspection and adjustment are no longer constant. It has been determined that the inspection time while using the testing machine is Normally distributed with a mean of 10 minutes and a standard deviation of 2 minutes. Inspection without the machine is Uniformly distributed between 2 and 8 minutes and adjustment time conforms to an Erlang distribution of order 2 with a mean of 45 minutes. The interarrival of TVs is Exponentially distributed with a mean of 12 minutes.

Figure 3 shows the modified network. The inspection activity has been divided into the nodes. Activity 3 represents the inspection with the testing machine and Activity 7 is the inspection without the testing machine. So that adjusted TVs can be logically distinguished from newly arrived TVs, Queue 6 has been added. Activity 3 now requires two resources, represented by the two columns of resource symbols. One resource from each column must be available before Activity 3 can begin. Symbols within one column specify alternative resources. In selecting an inspector at Activity 3, the newer inspector has 1st PRIority while the older inspector has 2nd priority. At the adjustment activity, the adjustor is given PRIority over the older inspector. At Activity 7 no queue is required for the TV because it simply retains the inspector from the CURrently completed inspection at Activity 3.

The older inspector may service TVs in Queues 2, 6, and 4. When the older inspector finishes an activity, TVs may be waiting in all three queues. Which one should be serviced? The decision is made by using a queue selector tree which describes the inspector's decision-making process. A selector node like node 30 is used to specify one decision. In this case, the older inspector PREfers to service TVs in Queue 6, then TVs in Queue 2, and lastly, TVs in Queue 4. The ability of resources to make decisions exerts a strong influence on the performance of a model and is a convenient modeling concept in INS.

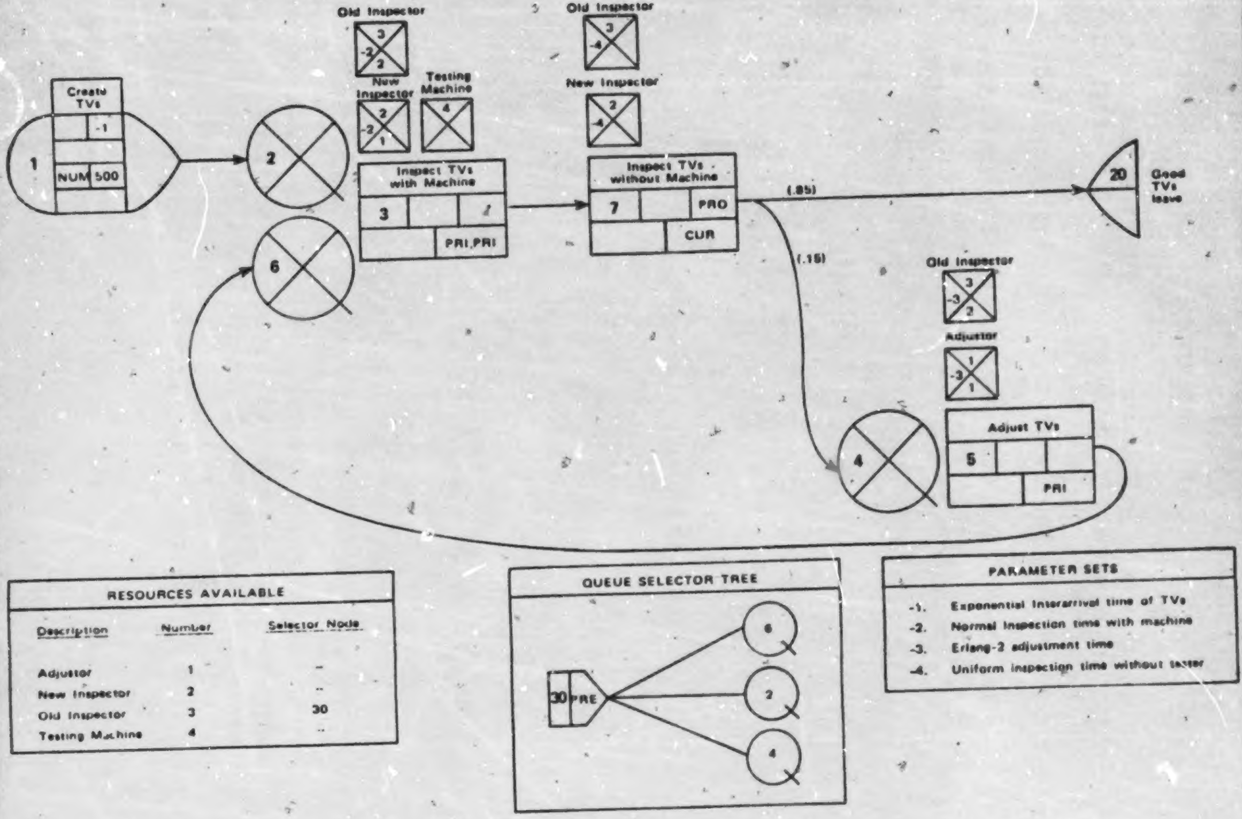
Finally, the constant times in the previous version have been replaced by negative numbers referring to parameter sets which specify random behavior according to a statistical distribution. INS automatically generates random deviates from a number of well known distributions and also allows the modeler to specify his or her own distribution.

Adding Individuality to the TVs

Transactions may not always represent undifferentiated units. We can distinguish transactions by assigning them attributes. For example, suppose there are two different types of TVs produced. Each type of TV requires a slightly different inspection time when using the circuit tester. Also, the testing machine is quite expensive and we wish to collect statistics on the utilization of the machine by the two types of TVs. The testing maching costs $300 per hour or $5 per minute. Finally, we wish to give TVs which have been adjusted more than once a higher priority at the inspection activity.

Figure 4 shows that a second source node has been added to model the second type of TV. TVs created at Source 1 are given the attribute TYPe = 1 while those created at Source 8 have TYPe = 2. A new node now appears called the assignment node. It has the following two parts:

Figure 3: Expanded TV Inspection and Adjustment Problem


Create TVs
1
-1
NUM 500
2
6
Old Inspector
3
-2
2
New Inspector
2
-2
1
Testing Machine
4
Inspect TVs with Machine
3
PRI,PRI
Old Inspector
3
-4
New Inspector
2
-4
Inspect TVs without Machine
7
PRO
CUR
(.85)
(.15)
20
Good TVs leave
Old Inspector
3
-3
2
Adjustor
1
-3
1
4
Adjust TVs
5
PRI
QUEUE SELECTOR TREE
30
PRE
6
2
4

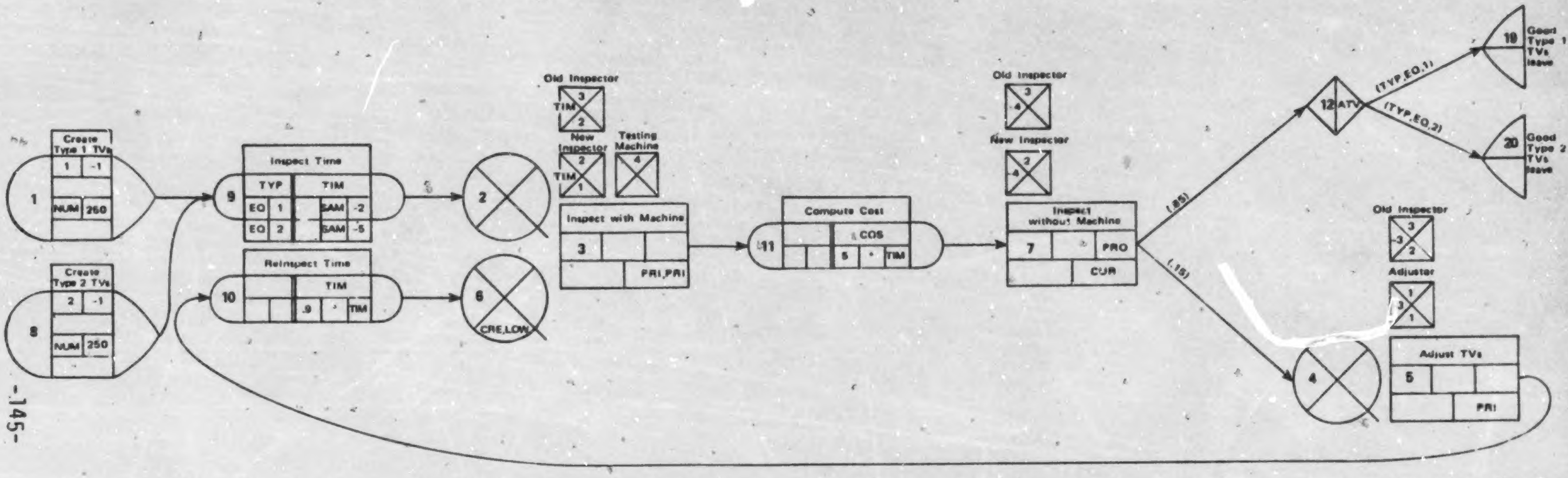
RESOURCES AVAILABLE

Description	Number	Selector Node
Adjustor	1	–
New Inspector	2	--
Old Inspector	3	30
Testing Machine	4	--

PARAMETER SETS

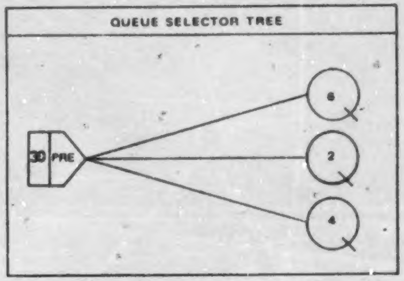
-1. Exponential Interarrival time of TVs
-2. Normal inspection time with machine
-3. Erlang-2 adjustment time
-4. Uniform inspection time without tester

Figure 4: Adding Individuality to the TVs



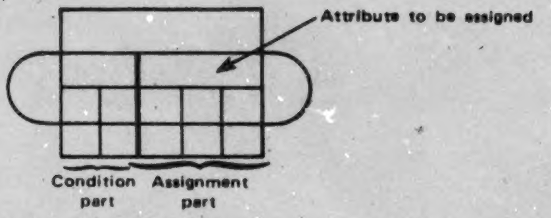
Create Type 1 TVs
1
-1
NUM
250
Create Type 2 TVs
8
2
-1
NUM
250
9
Inspect Time
TYP
TIM
EQ
1
SAM
-2
EQ
2
SAM
-5
10
Reinspect Time
TIM
.9
*
TIM
2
6
CRE,LOW
Old Inspector
3
TIM
2
New Inspector
2
TIM
1
Testing Machine
4
Inspect with Machine
3
PRI,PRI
Compute Cost
11
COS
5
*
TIM
Old Inspector
3
-4
New Inspector
2
-4
Inspect without Machine
7
PRO
CUR
(.85)
(.15)
12
ATV
(TYP,EQ,1)
(TYP,EQ,2)
19
Good Type 1 TVs leave
20
Good Type 2 TVs leave
Old Inspector
3
-3
2
Adjuster
1
-3
1
4
Adjust TVs
5
PRI

RESOURCES AVAILABLE		
Description	Number	Selector Node
Adjustor	1	–
New Inspector	2	–
Old Inspector	3	30
Testing Machine	4	–



QUEUE SELECTOR TREE
30
PRE
6
2
4

PARAMETER SETS	
-1.	Exponential interarrival time of TVs
-2.	Normal inspection time with machine (TYP-1)
-3.	Erlang-2 adjustment time
-4.	Uniform inspection time without tester
-5.	Normal inspection time with maching (TYP-2)



At Assignment 9, if TYPe EQuals 1, then TIMe is assigned a SAMple from Parameter set 2. If TYPe EQuals 2, then TIMe is assigned a SAMple from Parameter set 5. At Activity 3, we can see that TIMe appears in place of the -2, and thus INS will automatically use the TIMe obtained at Assignment 9.

Following inspection, we compute the COSt of using the machine as $5 per minute times the inspection TIMe (in minutes). By adding a TABLE statement (which can be seen in Figure 8), we can create a table containing the following information on machine cost: total cost; average cost per inspection; standard deviation, minimum, and maximum cost per inspection; and all of these statistics for each type of TV separately. Figure 5 contains the table produced on cost. After being adjusted, inspection time is recomputed as a fraction of the original time. At Queue 6, adjusted TVs are now ranked by LOWest CREation time, an attribute automatically assigned by INS when a transaction is created. This differs from the default ranking of first-in-first-out (FIFO) at Queues 2 and 4. A TV which has been adjusted more than once will be older than the other TVs and will be placed at the front of Queue 6.

Finally, good TVs enter a route node which enables transactions to make more than one decision in choosing a final destination. At Route 12, the TYPe 1 TVs branch to Sink 19 while TYPe 2 TVs branch to Sink 20. This allows us to collect statistics at the sink nodes for each type of TV.

Enlarging the Scope of the Model

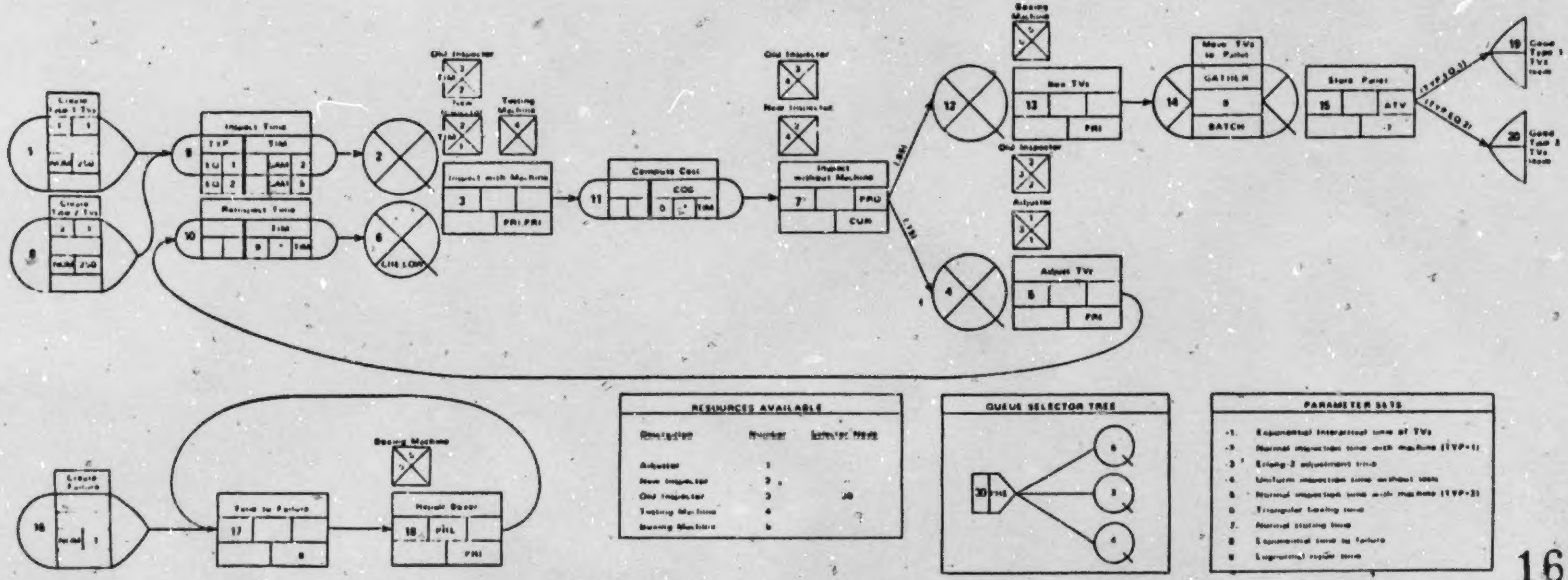
In our final enhancement, we wish to add to our model the following processes. After being successfully inspected, TVs are placed in boxes by an automated boxing machine. The boxer is subject to periodic breakdowns about every 100 minutes, Exponentially distributed. When the boxer breaks down, it is repaired while inspected TVs must wait for repairs to be completed. The repair requires a Lognoramlly distributed time having a mean of 20 and a standard deviation of 5 minutes. After being boxed, TVs are moved to a pallet. When the pallet has eight TVs, it is moved into a temporary storage area.

Figure 6 shows the additions made. At Activity 13, TVs are boxed and then moved to a pallet at Queue 14. At Queue 14, TVs are GATHERed until 8 have accumulated. Then all 8 TVs are stored together as a BATCH. Although no resources are required at Activity 15, Queue 14 is necessary because transactions must wait for other

Figure 5: Table of Testing Machine's Cost

```
                                   ** TABLE NUMBER    1 **

                                   COST OF TESTING MACHINE

                                    STANDARD                                  NUMBER OF                    SUM OF THE
TRANSACTION TYPE          MEAN      DEVIATION      MINIMUM        MAXIMUM     OBSERVATIONS        SUM        SQUARES

    1.00000             58.6484     15.8902        25.0000        100.000              62       3636.20      228660.
    2.00000             49.7066     11.5102        23.4000        75.0000              68       3380.05      176887.

     TOTAL              53.9712     14.4329        23.4000        100.000             130       7016.25      405547.
```

Figure 6: Boxing and Storing TVs



Old Inspector
Testing Machine
New Inspector
Boxing Machine
Inspect Time
Reinspect Time
Inspect with Machine
Compute Cost
COS
Inspect without Machine
Box TVs
Adjust TVs
Adjuster
GATHER
BATCH
Store Pallet
ATV
Good Type 1 TVs
Good Type 2 TVs
Create Failure
Time to Failure
Repair Boxer
RESOURCES AVAILABLE
Adjuster 1
New Inspector 2
Old Inspector 3
Testing Machine 4
Boxing Machine 5
QUEUE SELECTOR TREE
PARAMETER SETS
1. Exponential interarrival time of TVs
2. Normal inspection time with machine (TYP=1)
3. Erlang-2 adjustment time
4. Uniform inspection time without machine
5. Normal inspection time with machine (TYP=2)
6. Triangular boxing time
7. Normal storing time
8. Exponential time to failure
9. Lognormal repair time

transactions. This process is known as synchronization and gathering transactions is only one of several synchronization mechanisms available in INS.

To model the breakdown of the boxer, we add a second network. One transaction is created representing the failure of the boxer. After spending an Exponential amount of time at Activity 17, it enters Activity 18 which PREempts the boxing machine from Activity 13. The TV being boxed at Activity 13 is suspended until repairs are completed on the boxer (i.e. until Activity 18 ends). At that time, the boxer resumes boxing the TV and the failure transaction returns for another period of waiting at Activity 17.

INS OUTPUT

In a few pages we have presented a broad cross section of the modeling concepts available in INS. We have taken an elementary problem and expanded its scope in several directions with little difficulty. In addition to the ease with which modeling in INS is accomplished, INS automatically collects and reports a comprehensive set of statistics on all aspects of the model's behavior. Reports are produced on the number and time of transactions in the network, the number and time of transactions in all queues and activities, and the utilization of each resource. Also provided is information to assist the modeler in verifying the correct behavior of the network. For example, INS keeps track of the number of times each node is entered by transactions. A node which has never been entered frequently indicates an error in branching. The modeler can also request a trace of the transactions and resources moving through the network. Figure 7 shows part of the trace (generated by the MONITOR statement seen in Figure 8 which contains the entire set of statements needed to run the final version of the problem).

The statistics produced automatically by INS can be eliminated or enhanced by the modeler. We have already seen how INS can collect statistics on attributes. Additional statistics can also be collected on the nodes. Table 2 in Figure 8 collects statistics on the number of transactions in Queues 2 and 6 and Activities 3 and 7, i.e. the entire inspection subsystem. Figure 9 contains the histogram produced as part of Table 2. Other forms of statistics are also created just as easily providing the user with an ensemble of output capable of verifying the network's behavior, providing answers to questions about performance, productivity, and cost, and generating additional insight into the system being studied.

APPLICATION AND AVAILABILITY OF INS

INS has been applied in several environments. The 37 examples in the textbook (1) illustrate the range of applications for INS including queuing, inventory, computer, manufacturing, and service systems.

INS has been successfully used in simulation courses for engineering students. Graduate students utilized it as a new

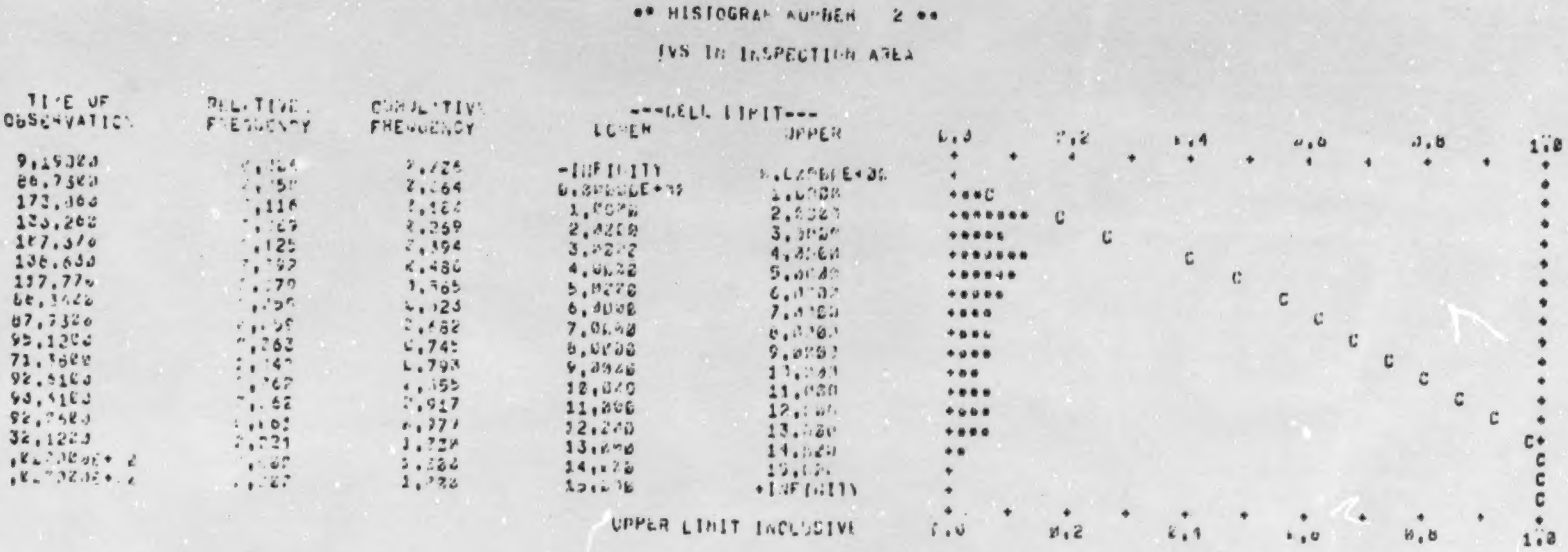
Figure 7: Tracing Transactions and Resources

```
   TIME      ENTITY  NUMBER     (TYPE)  ACTION               NODE TYPE  NUMBER                 RELATED INFORMATION

 321,350      TR.      23  (     2,00) COMPLETED     AT      ACTIVITY        3  ACTIVITY TIME =            16,440
              RES       1  (     2)    IN BUFFER STORAGE
              RES       4  (     3)    IN BUFFER STORAGE
              TR       23  (     2,00) LEAVING               ASSIGNMENT     11  INDIVIDUAL ATTRIBUTE CUS =  62,200
              RES       3  (     2)    BUSY          AT      ACTIVITY        7  ON TRANSACTION      23      WITH A RES ACTIVITY
              TR.      23  (     2,00) BEGINNING     AT      ACTIVITY        7
              TR.      24  (     2,00) DEPARTING             QUEUE           2  TIME IN QUEUE =            30,472
              RES       4  (     3)    BUSY          AT      ACTIVITY        3  ON TRANSACTION      24      WITH A RES ACTIVITY
              RES       2  (     2)    BUSY          AT      ACTIVITY        3  ON TRANSACTION      24      WITH A RES ACTIVITY
              TR.      24  (     2,00) BEGINNING     AT      ACTIVITY        3

 323,270      TR.      11  (     2,02) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           20
              TR.      12  (     1,20) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           19
              TR.      13  (     1,00) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           19
              TR.      15  (     1,60) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           19
              TR.      16  (     2,00) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           22
              TR.       9  (     2,00) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           20
              TR.      14  (     2,00) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           22
              TR.      17  (     1,00) COMPLETED     AT      ACTIVITY       15  ACTIVITY TIME =            55,650
                                       DESTROYED     AT      SINK           19

 365,670      TR.      29  (     1,00) CREATED       AT      SOURCE          1  APPOINTMENT TIME =      365,670
                                       LEAVING               ASSIGNMENT      9  INDIVIDUAL ATTRIBUTE TIN =  12,560
                                       ENTERING              QUEUE           2  THE NUMBER OF TRANSACTIONS WAITING IS           4

 365,750      TR.      23  (     2,00) COMPLETED     AT      ACTIVITY        7  ACTIVITY TIME =             5,400
              RES       3  (     2)    IN BUFFER STORAGE
              TR.      23  (     2,02) ENTERING              QUEUE          12  THE NUMBER OF TRANSACTIONS WAITING IS           1
              RES       3  (     2)    IDLE

 367,990      TR.      21  (     1,72) COMPLETED     AT      ACTIVITY       13  ACTIVITY TIME =            16,210
              RES       5  (     4)    IN BUFFER STORAGE
              TR.      21  (     1,70) ENTERING              QUEUE          14  THE NUMBER OF TRANSACTIONS WAITING IS           3
                                       GROUPED       AT      QUEUE          14  WITH TRANSACTION        16
              TR.      22  (     1,72) DEPARTING             QUEUE          12  TIME IN QUEUE =         16,210
              RES       5  (     4)    BUSY          AT      ACTIVITY       13  ON TRANSACTION      22      WITH A RES ACTIVITY
              TR.      22  (     1,20) BEGINNING     AT      ACTIVITY       13
```

Figure 8: Statements for the Final Version

```
$                  FINAL TV INSPECTION AND ADJUSTMENT PROBLEM
$
 GENERAL, INSPECT AND ADJUST TVS,1,(7)TIME TO STOP =,1500 $ MINUTES
 SET,(8)2
$         SET DISTRIBUTION  MEAN  MIN  MAX
 PARAMETER, 1, EXPONENTIAL ,,  23                $ TV INTERARRIVAL TIME
 PARAMETER, 2,   NORMAL    ,,  12,  5,  20, 3 $ INSPECT TIME W/ MACHINE--TYPE 1
 PARAMETER, 3,   ERLANG    ,,  45, 30,  70, 2 $ ADJUSTMENT TIME
 PARAMETER, 4,   UNIFORM   ,,    ,  2,   8    $ INSPECT TIME WITHOUT MACHINE
 PARAMETER, 5,   NORMAL    ,,  10,  5,  15, 2 $ INSPECT TIME W/ MACHINE--TYPE 2
 PARAMETER, 6, TRIANGULAR  ,,  10,  8,  12    $ BOXING TIME
 PARAMETER, 7,   NORMAL    ,,  50, 30,  70, 5 $ STORAGE TIME
 PARAMETER, 8, EXPONENTIAL ,, 100,            $ TIME TO FAILURE
 PARAMETER, 9,  LOGNORMAL  ,,  20,  5,  50, 5 $ REPAIR TIME
$
 TABLE, 1, COST OF TESTING MACHINE, A/T
 TABLE, 2, TVS IN INSPECTION AREA,, 15, 1, 1
$
 MONITOR,TRACE AT TIME ,300
 MONITOR,UNTRACE AT TIME ,330
$
 ATTRIBUTE,TIME                            $ INSPECT TIME WITH MACHINE
 ATTRIBUTE,COST,(6)1                       $ COST OF TESTING MACHINE - TABLE 1
$
$        NUMBER  TYPE  SELECTOR
 RESOURCE,   1,    1                       $ ADJUSTOR
 RESOURCE,   2,    2                       $ NEW INSPECTOR
 RESOURCE,   3,    2,     30               $ OLD INSPECTOR
 RESOURCE,   4,    3                       $ TESTING MACHINE
 RESOURCE,   5,    4                       $ BOXING MACHINE
$
 SELECTOR TREE FOR OLD INSPECTOR, 30, PREFERRED ORDER, 6, 2, 4
$
$******************************** THE NETWORK ********************************
$
 SOURCE,1,,1,-1,,,NUMBER TO CREATE =,250 $ CREATE TYPE 1 TV SETS
    BRANCH TO, 9
 SOURCE,8,,2,-1,,,NUMBER TO CREATE =,250 $ CREATE TYPE 2 TV SETS
$
 ASSIGNMENT, 9,, TYPE ,       TIME =   ,
                 EQ, 1,    , SAMPLE, -2,
                 EQ, 2,    , SAMPLE, -5  $ ASSIGN INSPECTION TIME BY TV TYPE
$
 QUEUE,2,2                                 $ QUEUE OF ARRIVING TVS
 QUEUE,6,2,(6)CREATION TIME,LOWEST FIRST $ QUEUE OF ADJUSTED TVS
 ACTIVITY,3,,2,(8)PRIORITY,,PRIORITY      $ INSPECT TVS WITH MACHINE
    GROUP,1, 2,,1,TIME, 3,,2,TIME         $ BY AN INSPECTOR
    GROUP,2, 4                             $ USING THE TESTING MACHINE
$
 ASSIGNMENT,11,,,COST =,,,5,*,TIME        $ COMPUTE COST AT $5 PER MINUTE
 ACTIVITY,7,PRO,2,(8)CURRENT               $ INSPECT TVS WITHOUT MACHINE
    GROUP,1, 2,,,-4, 3,,,-4                $ BY THE SAME INSPECTOR
    BRANCH TO, 12, .85                     $ 85 PERCENT PASS
    BRANCH TO,  4, .15                     $ 15 PERCENT FAIL
$
 QUEUE,4
 ACTIVITY,5,(8)PRIORITY                    $ ADJUST TVS
    GROUP,1, 1,,1,-3, 3,,2,-3              $ BY THE ADJUSTOR OR OLD INSPECTOR
 ASSIGNMENT,10,,,TIME =,,, .9,*,TIM        $ ASSIGN REINSPECTION TIME
    BRANCH TO, 6                           $ RETURN FOR REINSPECTION
$
 QUEUE,12
 ACTIVITY,13,(8)PRIORITY                   $ BOX TVS
    GROUP,1, 5,,,-6                        $ BY THE BOXING MACHINE
$
 QUEUE,14                                  $ MOVE TVS TO PALLET
    SYNCHRONIZE,GATHER,8,BATCH             $ WITH 8 TVS PER PALLET
 ACTIVITY,15,ATV,(8)-7                     $ STORE PALLET
    BRANCH TO, 19, TYPE,EQ,1
    BRANCH TO, 20, TYPE,EQ,2
$
 SINK,19                                   $ GOOD TYPE 1 TVS LEAVE
 SINK,20                                   $ GOOD TYPE 2 TVS LEAVE
$
$************** BOXING MACHINE FAILURE LOOP **************
$
 SOURCE,16,,1                              $ CREATE FAILURE
$
 ACTIVITY,17,(8)-8                         $ EXPONENTIAL TIME TO NEXT FAILURE
$
 ACTIVITY,18,(6)PREEMPT,,PRIORITY          $ REPAIR BOXING MACHINE
    GROUP,1, 5,,,-9                        $ USING THE BOXING MACHINE
    BRANCH TO, 17                          $ LOOP BACK FOR MORE FAILURES
FINISH
```

Figure 9: Distribution of the Number of TVs in the Inspection Area



```
                                   ** HISTOGRAM NUMBER   2 **

                                   TVS IN INSPECTION AREA

  TIME OF       RELATIVE     CUMULATIVE       ---CELL LIMIT---
OBSERVATION    FREQUENCY     FREQUENCY     LOWER          UPPER         0.0      0.2      0.4      0.6      0.8      1.0
                                                                        +    +    +    +    +    +    +    +    +    +
  9.19320         .064          .026     -INFINITY     0.12000E+00     +                                            +
 86.7500          .058          .064     0.12000E+00    1.0000          +**C                                         +
173.460           .116          .184      1.0000        2.0000          +*******  C                                  +
125.260           .069          .269      2.0000        3.0000          +****      C                                 +
187.370           .125          .394      3.0000        4.0000          +*******      C                              +
136.630           .092          .486      4.0000        5.0000          +*****+        C                           +
137.770           .079          .565      5.0000        6.0000          +*****           C                         +
 66.3420          .059          .623      6.0000        7.0000          +***               C                       +
 87.7320          .059          .682      7.0000        8.0000          +***                 C                     +
 95.1200          .063          .745      8.0000        9.0000          +***                   C                   +
 71.3600          .047          .793      9.0000       10.000           +**                      C                 +
 92.5100          .062          .855     10.000        11.000           +***                       C               +
 93.5100          .062          .917     11.000        12.000           +***                         C             +
 92.7500          .060          .977     12.000        13.000           +***                           C           +
 32.1220          .021          .998     13.000        14.000           +*                                         C+
  .[illegible]E+ 2   .007      [illegible] 14.000       15.000           +                                          C
  .[illegible]E+ 2   .007       1.000    15.000       +INFINITY         +                                          C
                                                                                                                   C
                                                                                                                   C
                                                                        +    +    +    +    +    +    +    +    +    +
                                          UPPER LIMIT INCLUSIVE         0.0      0.2      0.4      0.6      0.8      1.0
```

network-based language while it has been used as the principle simulation methodology in an undergraduate industrial engineering course in systems analysis. The language has also been used as the basis of a senior elective in medical school. Medical students have used INS to create models for clinical investigation. These instances of its use in a teaching environment have demonstrated that, not only can the language be quickly learned and applied by persons with minimal technical expertise, but it can also be valuable to those with deeper appreciation of simulation.

INS is written in ANSI FORTRAN making it portable and has been implemented on a number of machines. In particular, INS is being used on the DECsystem-10 at Indianapolis and the CDC 6600 at Bloomington. INS and the textbook are available from the Regenstrief Institute, 1001 West Tenth Street, Indianapolis, Indiana, 46202. The text, Simulation Modeling With INS, is available for $15. INS is actively being maintained by the Institute and is the subject of continuing improvement both to increase the efficiency of its execution and to extend its applications. For further information you can call (317) 264-4566.

CONCLUSION

Because simulation modeling is a process of successive refinement, there is a need to be able to quickly produce a working "first cut" model and to embellish this model as greater insight is generated. INS facilitates model building by providing a simple, but powerful set of symbols which can be quickly combined to create a network model of a system. By a simple translation of the INS network to INS statements, INS conducts the simulation. It relieves the modeler from the burdens of simulation mechanics and programming. The approach allows the modeler to focus on the important tasks of model creation and validation. INS makes embellishments to the model easy because they can be accomplished by changing a specification at a node or altering some segment of the network. INS is easy to learn and use, not only by making it easy to create a simulation model but by giving structure to the modeling process. Since INS is visually appealing and easily documented, it is an excellent vehicle for communication between the modeler and the client, encouraging cooperative participation. Last but not least, INS is written in ANSI standard FORTRAN and is available for a variety of computers.

REFERENCES

1. Roberts, S.D., and J.S. Schier, Simulation Modeling With INS, Regenstrief Institute for Health Care, Indianapolis, Indiana, 1979.

Simulating American Social Structure:
An Interactive Game-Simulation for Career and Life Cycle Decisions

Scott C. Shreiner
Assistant Professor of Sociology
Indiana University-Purdue University at Fort Wayne

I. Introduction

The use of computerized game-simulations as a research method and as an instructional device for the classroom setting derives from the fundamental assumptions that: (1) games are an almost universal aspect of human existence and (2) the structure of games is isomorphic to the normative aspects of social structure (Coleman 1969). These assumptions provide the basis for paradigm development, investigating individual behavior in social situations whose essential features are simulated within the game. From this assumption that social structures can be created in game form follows the premise that the problems faced by people in playing such games is congruent with the essential aspects of problems that they would or could face in real life. These assumptions also formed the motive base for the development of the CALIS (Career and Life Interactive Simulation) game-simulation.

The CALIS simulate is the third in a line of career oriented games (Boocock, 1968; Tallman and Wilson, 1975). Unlike its predecessors, CALIS is not a board game controlled by experimental assistants basing its reward structure on points earned during the game. The problems of this format (Tallman and Wilson, 1974) are not inherent in the computer interactive form of this game-simulate. CALIS is perhaps the most complex and sophisticated instrument for the analysis of player decision choices in the career and life cycle area.

II. Purpose for Developing CALIS

The CALIS computer based game-simulation was developed as the major research instrument for the Occupational Choice and Differentiation of Career Paths project conducted by the author, Dr. Robert K. Leik of the University of Minnesota and Dr. Anthony Harris of the University of Massachusetts.

Originally, CALIS was developed to investigate the differential entry of blacks and other minorities into the labor market. The complexity of CALIS grew out of the research and a counseling component inherent in this research. The research component utilized CALIS to investigate the "motivational deficiency" hypothesis that blacks and other minorities are not sufficiently motivated into high status occupational positions due to the influence of subcultural norms mediating against this status attainment. The projects alternative hypothesis was that blacks and minorities suffered not from motivational deficiency but rather from a lack of knowledge about career pathways, the occupational reward structures and the consequences of immediate versus deferred gratification. The decision making areas to be presented shortly were alligned to tap into the complexity of these hypothesis. The career counseling component grew from the alternative hypothesis. If a simple lack of knowledge of career paths and life cycle decisions was found to be the dominant cause of differential entry into the labor market, then the CALIS game-simulate could be used to aid minorities in discovering the ways to higher status positions. By playing this computer game-simulation with a counselor a person could discuss their playing patterns and the isomorphism of these patterns with their real life behaviors and alternatives to their current life styles could be explored with the game-simulation assessing the consequences of the players decisions.

III. CALIS: A Description

CALIS was developed to function as a non-discriminating social structure with two major tasks: (1) to present decisional alternatives to the players in major aspects of life and (2) based on these decisions evaluate the performance of the player in terms of a series of standards and communicate to the player the consequences associated with these decisions. CALIS is non-discriminating in the sense that regardless of a players race, sex, ethnicity or social class the same evaluation standards are applied to each player. In other words the game-simulate hold what are considered sociologically as structural determinants of life chances constant for all players. These standards are based on a series of optimum resource allocations developed by the researchers and in part derived from the empirical sociological literature.

The CALIS game-simulation places the player in the social-psychological situation of an individual confronting American society during the late adolescent through early middle age stages of the life cycle (from age 17 to age 35). This social-psychological situation is simulate-operationalized as a micro-economic decision making process where a player allocates his or her time and money to various aspects of the game. The mode of play permits the player to allocate these resources in any "life area" allowing the players to determine the importance of each area and the order of play. This permits players to develop their own life styles and change them during the course of play. Beyond the resource allocation consideration, CALIS also collects player self-evaluations of the utility of their major decisions and the probability of success of attaining their game-oriented goals. The computer uses a graphic ratio-to-standard measure to make these utility and probability determinations.

In terms of game-consequences, CALIS evaluates a players performance and provides feedback based on these evaluations. Resources allocated in the education area determine a round by round grade for the player in high school, college or professional school. These evaluations then cumulate and determine a players successful application to college and later in the game to graduate or professional school. These educational evaluations also govern the players ability to transfer to different schools. The final educational evaluation also forms the basis for the transferability of performance rating from the education to the occupation life area.

Within the occupational area, similar performance ratings on the players are generated, again based on resource evaluations. These evaluations then determine such things as pay increases, job promotions and the ability for a player to transfer from one job to another.

The third area of evaluation utilizes a concept of life balance. This concept assumes that if a person is to live a non-pathological life, the players must balance their time between work or education and social support activities (e.g., family, friends). Where a player devotes an excessive amount of time to one area of play to the neglect of the other areas, CALIS provides the player with negative feedback messages and if the balance is not changed the players performance ratings in other areas will begin to be negatively effected.

In sum, the social structure of the CALIS game-simulation performs a continuous process of player evaluations, just as society through its agencies evaluates each individual continuously. The consequences of these evaluations may either open or close opportunities and career paths for an individual player as well as determining simulate payoffs to the player.

One final note on the continuous processes in CALIS. The game simulate also provides the player with a running, "on-demand" accounting facility which feeds back to the player the current amounts of time and money remaining to be allocated in the round. At the end of a round (one round equals 1 year) any hours not allocated are lost to the player and any money remaining is automatically transferred to the players savings account.

A. The Life Areas of CALIS

As in any game simulation the realism of the simulation structure depends on its designers understanding of the system to be modeled and the number of causal linkages reflected in the model. When CALIS was created it was recognized that only by providing the game with a maximum of flexibility could the realism of social life be captured. In other words, the decision making process had to be such that free access to all game areas be given to the player in a round of play. CALIS was designed in a multilateral way rather than in a linear manner which would have forced the subjects to play the life area in sequence. Figure 1 diagrams the overlay flowchart which shows how this flexibility was accomplished.

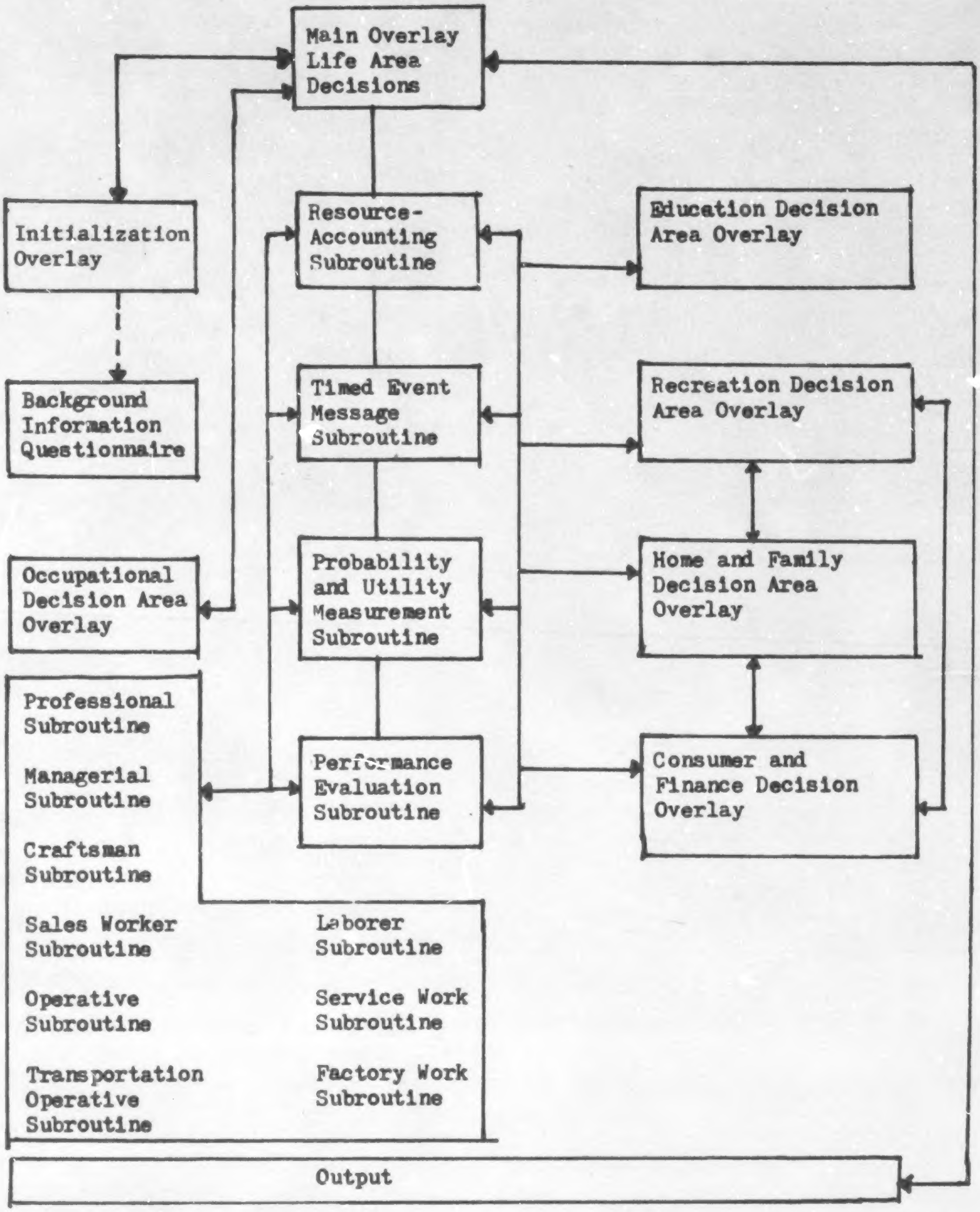

Main Overlay Life Area Decisions
Initialization Overlay
Background Information Questionnaire
Occupational Decision Area Overlay
Resource-Accounting Subroutine
Timed Event Message Subroutine
Probability and Utility Measurement Subroutine
Performance Evaluation Subroutine
Education Decision Area Overlay
Recreation Decision Area Overlay
Home and Family Decision Area Overlay
Consumer and Finance Decision Overlay
Professional Subroutine
Managerial Subroutine
Craftsman Subroutine
Sales Worker Subroutine
Operative Subroutine
Transportation Operative Subroutine
Laborer Subroutine
Service Work Subroutine
Factory Work Subroutine
Output

Figure 1. Overlay Flowchart of the CALIS Game Simulation

This mode of game structuring allowed for the players to create their own career paths. A few of the many life cycle and career paths that emerged during the piloting of CALIS are:

(1) High School → College → Graduate School →
Occupation → Marriage

(2) High School → Marriage → Occupation →
College —— New Occupation

(3) High School (Drop out) → Work → Marriage →
Divorce

(4) High School → College (Drop out) → Marriage →
Work

These paths reflect a wide variety of high school students perceptions as to the timing of significant life cycle events and entry into the labor market.

Not only was CALIS designed to discover over time career path sequencing but also to probe within round patterns of decision choice, in terms of player preference for the various life areas. A variety of within round possibilities of area importance also emerged during piloting, for example:

(1) Recreation area --- Education area --- Finances area

(2) Education area --- Part time employment --- Recreation area

(3) Employment area --- Education area --- Recreation area --- Finances area

In the post game interview with the pilot subjects, it was learned that these sequences tended to indicate those areas of the game which were most important to them or most enjoyable.

1. The Educational Decision Making Area

CALIS assumes that the player is about age 17 as the game begins. As such the player is assumed to be a high school junior. After entering the education decision area the player indicates whether or not he or she is in an academic or a vocational track. Then the player allocates time for studying per week for the round. The subject may also change from one curriculum to the other if their performance evaluation in round 2 is high enough. During the players senior high school round this person may decide to apply to several types of colleges or technical schools, regardless of whether or not the computer-counselor advises this decision or not based on the player's evaluation. College options include Ivy League college, attending a state university, a community college or a technical school. The player is given a time-delayed feedback message as to his or her acceptance or rejection by these schools. Each type of school has a different level of evaluated performance to rate the player for acceptance.

If accepted, the college cycle permits the player to pick a "major" signified in terms of professional careers which match those in the occupational routine exactly. The decision of a major, divides the players performance evaluations into "hard" and "soft" sciences and professions. This division differentially weights the performance evaluation in terms of optimum study time allocations. Besides, the normal study time decisions, a player may also transfer from one type of college to another, again depending on their performance rating.

If a person chooses a college rather than a technical school, during their senior year a player may be advised to apply to graduate or professional school. Again the player goes through the application process, the evaluation process and the acceptance-rejection process. The types of graduate education include, Ivy League graduate school, state university graduate study and professional school (e.g., law school or medical school). The remaining decisions are again similar to the college routine but again the optimum allocation of studying time increases, to remain in school.

After a player completes the education cycle, stopping or restarting this cycle at will during the game, CALIS creates a transfer score which is used in the application process for entry into the labor market.

2. The Labor Market or Occupational Decision Making Area

The occupational decision making area included nine sub-area of jobs which correspond to representative examples of positions from the major classifications from the 1970 Census.

Prior to a player terminating their educational cycle, he or she is limited only to part-time work at up to 30 hrs./week. Once out of school, for whatever reason, a player is permitted to hold a full-time job. Part-time work is limited to only jobs in the Sales, Clerical, Craftsman, Operative, Laborer and Service categories. All part-time jobs have a starting salary of $3.00/hr. Players work for the number of hours of their choice. Applications for part-time work are evaluated on a random basis where the player has a 60% chance of getting the part-time work. This procedure is meant to simulate a job market where there is a large number of applicants for any job with each being about equally qualified since the majority of these positions do not have a high skill level. Again, a player may change part-time jobs and hold more than one job at a time.

For a full-time job, the player must be out of school during the round that he or she applies for the job. Later, after obtaining the job, the player can reenter school on either a part- or full-time basis.

The offers of starting salaries for the full-time positions were based on median incomes for each job as reported in the 1970 census. CALIS also assigns vacation time and work hours per occupations with the exception of professional and managerial positions. For these positions a player is expected to spend extra hours working in order to obtain salary increases and promotions.

Obtaining each position naturally involved an application process and a player performance evaluation based on college performance, whether or not a player's college major has prepared this person for the position (if this is the first job application), previous job performance evaluations and a random component again representing job competition. If CALIS makes a job offer, it still rests with the player to make the acceptance decision based on the job description furnished by the computer.

After a full-time job is accepted, a player may choose to change jobs in the same category or into another job category. A player may easily exhibit downward mobility by applying for a lower status position. Upward mobility is only accomplished through returning to the education process. Again transferability of job experience is based on job performance which effects the application process and the new starting salary.

Finally, a player holding a full-time job may also hold a part-time position which will increase that players game-income and only has an effect on performance evaluations through the life balance component.

3. The Recreational Decision Making Area

The recreational decision making area in conjunction with the consumer-financial area was developed to investigate the players conceptions of deferred or immediate gratification. This area provides a wide range of activities over which a player may allocate time and money. These activities include sports, friendships, visiting relatives, watching television, reading, hobby-craft activities, music related activities, church related activities, vacations and dating behaviors. Instead of the game presenting specific activities in these areas, CALIS asks open ended questions and permits the player to describe their own recreational activities in detail. This procedure adds to the realism of the game and avoids any implicit sexism or racism in the nature of the activities.

In activities where the purchase of equipment is necessary in the game situation, a player is transferred to the consumer-finance area for the purchase of this equipment. Dating activities are a necessary prerequisite for marriage in the Home and Family area. Visiting with friends and relatives enters the evaluation component of the game since time spent on these activities contributes to the life balance evaluation.

4. The Home and Family Decision Making Area

This area of CALIS is concerned with several critical life cycle events. Primarily these events include breaking with ones family of orientation and starting ones family of procreation. This area allows for decisions necessary for the development of independent living, renting an apartment or home, home ownership, getting married, having children, the option of getting a divorce, and moving the family unit. Time and money allocations are again necessary for each decision and the amount of time that a player spends with their "family" contributes to the life balance performance evaluation.

5. The Consumer-Finances Decision Making Area

This final area of the CALIS game-simulation was developed in conjunction with the recreation area to investigate the deferred versus immediate gratification component of the major hypothesis. This area explores consumer behavior by allowing the player to make purchases of clothing, sports equipment, automobiles, boats, furniture, home repairs, books, magazines, televisions, hobby equipment, records and tapes, alcohol and cigarettes. Once purchased the game also allows for these commodities to be sold at a fraction of their original purchase price.

As a sub-component of this area a player may also take out loans, that are evaluated on the basis for work performance, and open a savings account. Unlike the recreation area where most of the resource allocations are made in terms of time, the finances area is mainly money oriented in its allocations.

IV. Validity of the CALIS Game-Simulation

Given the complexity and multiple purposes of this game simulation, the usefulness of standard criteria of validity appears questionable (Snubik, 1970). However, two criteria appear relevant at this stage in the development of CALIS; face validity and rule consistence.

Face validity describes the extent to which the game reproduces the simulandum (Boocock, 1972; Hermann, 1967). In this case the simulandum is American social structure as viewed from the perspective of the individual. Boocock (1972) notes that it is imperative that the player view the game-simulation as being consistent with their experience in the real world.

To assess this aspect of validity, two small groups of pilot subjects were used and asked to evaluate CALIS from their own experiences. One group was an older group of university staff (ages 25-40) who were asked to attempt to reconstruct their life experiences using CALIS. The second group of players were high school juniors from Easthampton High School in Massachusetts. This latter group was asked to evaluate the game in terms of their expectations for the next ten years.

In the older group, seven out of ten players were able to reconstruct their own career paths to their current age. If their age was less than the maximum simulate age, these players also found that the game met their expectations for the next years. The three players who were not able to reconstruct their lives completely, pointed out that some major and unforseen event had occurred to them at some point in their careers and that the impact of this event changed their career paths. At present such events are not included as part of CALIS. In the high school group, their comments during post play inverviews indicated that all aspects of their current lives were included as part of the game decisions (i.e., all legal aspects) and that play of the game conformed to some of their conceptions of their futures. Again the presence of a major event occurring in the life of one of these players prevented him from setting up a full description of his life decisions. These students also commented that while future play conformed to some of their expectations, they indicated that they did not realize how complex their "future" lives could become. Two other comments from these pilot players of note: (1) that the playing time required for the game (2-3 hr.) did not produce fatigue but rather kept the students at a high level of stimulation and (2) that each had learned something from the game and desired to try out new simulate life styles at a later time. From the comments of these pilot players during the post play interview sessions it appears that CALIS has reasonable face validity.

The second aspect of validity relevant to CALIS is rule consistency. This aspect of validity is concerned with the consistent application of the game rules to all players. In the board versions of career games this is a problem originating with the research assistants and the differential application of rules (Tallman and Wilson, 1974). With the computer game-simulation there is little problem with the application of rules. As stated before, CALIS was conceived to be non-discriminating and all rules are applied consistently to all participants. The computerization of this career simulation has resolved all of the rule consistency problems which plague board games. Overall, at this stage in the development, CALIS has good validity by these criteria. Further experimentation in full research settings would enable the game to be evaluated for discriminant or predictive validity.

V. Conclusion: The Future Utility of CALIS

This paper has presented a description of the CALIS career and life cycle game simulation. The potential utility of this simulation remains to be fully explored. In these last few paragraphs, I will suggest some of the possible ways in which CALIS may be utilized.

In terms of the game's utility as a research instrument, the pilot study found several different career paths. Many more remain to be uncovered and explored in terms of their subcultural linkages. The complexity and amount of information provided by CALIS would permit many different kinds of research endeavor. Such endeavors could include further pursuit of the motivational vs knowledge questions which prompted the development of CALIS, the exploration of sex differences in career paths, studies of future consumer behavior, life cycle research exploring such significant events as marriage and desired family size, and labor market studies. With minor modifications CALIS has the potential of starting play at any age or stage in the life cycle for the players. Similarly, the duration of play could be expanded and studies of middle and old age decisions could be explored in the simulation.

Beyond its potential as a research instrument, CALIS also shows promise as a career counseling and guidance device. As our pilot study indicated, the players desired to utilize CALIS as a "safe" way of exploring different life styles and their consequences. In the counseling situation, sexual, racial, and ethnic patterns of play may reveal problems that a person may encounter or self-imposed barriers to achievement which may be rectified. A report summary from CALIS, would enable a counselor to examine, in detail, the career expectations of a player.

In summary, CALIS is a game-simulation waiting to be utilized to explore some of the more complex aspects of life cycle and career patterns on an individual or group level.

Technical Notes on CALIS

CALIS is written in FORTRAN for the C.D.C. Cyber 74 facility at the University of Massachusetts at Amherst. It is currently being modified for use in the Indiana C.D.C. Network. The program is in overlay structure and requires 54k to compile and 48k to load. For use on the Indiana system, this requires separate compilation and loading operations.

The game simulation is meant to be played interactively on any 16 line CRT terminal. Input is freefield from the terminal. CALIS creates two types of output--a round by round accounting of all decisions made in the game and a restart file. The restart file enables the researcher to restart a player at the (n-1)th round of play if the game has not been completed.

The game also includes a series of control switches that may be set by the researcher. These control switches may selectively cut off portions of the game so that only portions of the game may be used and tailored to the needs of the project.

The following three pages illustrate play of the CALIS game in each of the decision areas.

SAMPLE OF PLAY FOR CALIS

```
 IS THIS A RESTART ? 1=YES, 0=NO
? 0
 IS THIS A PRACTICE ROUND? 1=YES 0=NO
? 0
```

Option permitting a restart at any round

Option for practice or play

```
        WELCOME
 YOU ARE ABOUT TO PLAY A GAME THAT      PERMITS YOU
 TO MAKE DECISIONS ABOUT HOW TO SPEND TIME AND      MONEY
 IN A SIMULATED CAREER

 NOTE   THIS IS NOT A TEST     THERE ARE NO WRONG CHOICES.
 IF AT ANY TIME YOU WANT TO STOP     THIS GAME AT ANY DECISION-TYPE 999
 NOTE  ANY QUESTION THAT     ASKS A YES/NO QUESTION 1=YES 0=NO ALWAYS
 AFTER A QUESTION      HAS BEEN ASKED PLEASE WAIT UNTIL THE COMPUTER PRINTS A  ?
 IF YOU RESPOND TOO FAST THE COMPUTER WILL STOP THE GAME.
 THE ASSISTANT WILL GIVE YOU ANY FURTHER INSTRUCTIONS.
                 GOOD LUCK
? 0
```

Computer Introduction and Instructions to player-subject

```
 PLEASE ANSWER THE FOLLOWING QUESTIONS ABOUT YOURSELF
 AND YOUR FAMILY. ALL ANSWERS ARE TOTALLY CONFIDENTIAL
 TYPE IN A ZERO IF YOU DON^T KNOW AN ANSWER OR DO NOT WISH TO
 TO RESPOND.
 WHAT IS YOUR PRESENT AGE,PLEASE?
? 17
 WHAT IS YOUR SEX? 1=MALE 0=FEMALE
? 1
 WHAT IS YOUR RACIAL OR ETHNIC BACKGROUND?
 1=BLACK AMERICAN
 2=NATIVE AMERICAN
 3=WHITE     AMERICAN
 4=HISPANIC AMERICAN
 5=ASIAN AMERICAN
? 3
 HOW MANY BRoTHERS AND SISTERS WERE/ARE IN YOUR FAMILY?
? 1
 ARE YOU PRESENTLY ?
 1=SINGLE (LIVING WITH PARENTS
   2=SINGLE (LIVING ALONE)
 3=MARRIED
 4=WIDOWED
   5=SEPARATED
 6=DIVORCED
? 1
 ARE/WERE YOUR PARENTS?
 1=MARRIED AND LIVING TOGETHER
 2=SEPARATED (YOU LIVED WITH YOUR MOTHER)
 3=SEPARATED (YOU LIVED WITH YOUR FATHER)
 4=DIVORCED (YOU LIVED WITH YOUR MOTHER)
 5=DIVORCED (YOU LIVED WITH YOUR FATHER)
 6=WIDOWED (YOU LIVED WITH YOUR MOTHER)
 7=WIDOWED (YOU LIVED WITH YOUR FATHER)
 8=BOTH PATENTS DECEASED,YOU LIVED WITH A RELATIVE
 9= RAISED IN A CHILDRENS HOME OR INSTITUTION
? 1
```

Computerized Questionnaire (currently for information purposes only but may be modified to begin play at any point in a players life cycle)

```
 FROM THE LIST OF OCCUPATIONS, PLEASE FIND AND TYPE
  IN THE NUMBER OF THE OCCUPATION WHICH IS CLOSEST TO YOUR FATHERS  JOB.
? 235

 FROM THE LIST OF OCCUPATIONS, PLEASE FIND AND TYPE
  IN THE NUMBER OF THE OCCUPATION WHICH IS CLOSEST TO YOUR MOTHE'S  JOB.
? 997
```

Occupation numbers correspond to categories in the 1970 U.S. Census occupation dictionary

```
 PLEASE PICK THE CATEGORY WHICH COMES THE CLOSEST TO
 DESCRIBING YOUR FAMILY INCOME FOR THE LAST YEAR.
 1=UNDER $5,000
 2=$5,000 TO $10,000
  3=$10,000 TO $15,000
 4=$15,000 TO $20,000
 5=$20,000        TO $20,000
  6=$25,000 TO $30,000
 7= OVER $30,000
? 3
 PLEASE ENTER THE NUMBER OF YEARS OF YOUR FATHERS SCHOOL ING.
? 8
 PLEASE ENTER THE NUMBER OF YEARS OF YOUR MOTHERS SCHOOL ING.
? 12
 WHAT ar$K=UR PRESENT EDUCATIONAL STATUS?
   1=HIGH SCHOOL-COLLEGE PREPARATRY
   2=HIGHSCHOOL-VOCATIONAL TRAINING
   3=IVY LEAGUE COLLEGE
   4=STATE UNIVERSITY
   5=JUNIOR OR COMMUNITY COLLEGE
   6=TECHNICAL COLLEGE
   7=EXTENSION EDUCATION
   8=IVY LEAGUE-GRADUATE SCHOOL
   9=STATE UNIV-GRADUATE SCHOOL
  10=IVY LEAGUE-PROFESSIONAL SCHOOL
  11=STATE UNIV-PROFESSIONAL SCHOOL
  12=NOT ATTENDING SCHOOL
? 1
SINCE THE GAME PROVIDES FOR YOU TO SET UP YOUR PRESENT
 OCCUPATIONAL STATUS,NO QUESTIONS WILL BE ASKED ABOUT ANY
  JOB THAT YOU MAY HAVE NOW.
```

SAMPLE OF PLAY FOR CALIS

I. LIFE GOALS AND PERCEIVED PATHWAYS ORIENTATIONS

```
   "
 @ YOU ARENOW STARTING ROUND   1 YOUR AGE IS   17
  YOU MADE        0.00 DOLLARS FOR PLAYING THIS GAME
   "
  BY ORDER OF THEIR  IMPORTANCE TO YOU
  PLEASE RANK      THESE MAIN GOALS IN LIFE.
      1= A HIGH PAYING JOB
      2= A HAPPY MARRIAGE
      3= BEING WELL LIKED BY FRIENDS AND FELLOW WORKERS
      4= HELPING OR SERVING OTHERS
      5= BEING AN IMPORTANT OR POWERFUL PERSON
 ? 1,2,3,4,5
```

LIFE GOALS

```
  WHAT DO YOU THINK IS THE BEST PATHWAY TO ACHEVING       THE FIRST GOAL?
  PLEASE NUMBER THE FOLLOWING STEPS TO THIS      GOAL.
     1= GET A COLLEGE EDUCATION
     2= GET JOB EXPERIENCE
     3= GET MARRIED AND HAVE CHILDREN
     4= BUY EXPENSIVE THINGS
     5= MAKE AND KEEP MANY FRIENDS
 ? 1,2,3,4,5
```

PATHWAY TO FIRST LIFE GOAL

```
  YOU PRESENTLY HAVE THE FOLLOWING RESOURCES REMAINING THIS YEAR
  NUMBER OF HOURS =           4000 (    76 /WEEK)
  NUMBER OF DOLLARS = $   1000.00 (    19.23 /WEEK)
  AMOUNT IN SAVINGS ACCOUNT = $        0.00
  AMOUNT IN DEBT      = $        0.00
 ? 1
   "
```

Automatic resourse check

```
  THE FOLLOWING ARE LIFE AREAS WHERE YOU CAN MAKE ONE OR     MORE DECISIONS
  1= EDUCATION
  2 = RECREATION
  3 = EMPLOYMENT
  4 = HOME    AND FAMILY
  5 = FINANCES
  6 = END THE ROUND
  7 = END THE GAME

  IN WHICH LIFE AREA WOULD YOU LIKE TO MAKE DECISIONS?
 ? 1
   "
```

Life area decision choices

II. THE EDUCATIONAL DECISION AREA

Sample of play from junior year in high school only.

```
  WHAT IS YOUR PRESENT CURRICULUM
  1=COLLEGE PREPATORY
  2=VOCATIONAL CURRICULUM ?
 ? 1
   "
  HIGH SCHOOL COURSES ARE 5/TERM
  YOU SPEND 6HR/DAY      IN CLASSES.
  HOW MANY HOURS/DAY DO YOU WANT TO SPEND STUDYING FOR
 THESE COURSES ? (INCLUDING INSCHOOL AND HOME STUDY)
 ? 4
```

Time resourses allocated for grade determination

```
  HOW SATISFYING IS A GOOD GRADE TO YOU?
   --------------------
 ? ----------------------------------
  CHANCES OF GETTING SATISFACTION
   -----------------
 ? ----------------------------------
   "
  YOU WILL BE NOTIFIED LATER ABOUT YOUR GRADES
          FOR THIS YEAR.
```

Truncated ratio-to-standard measure for utility (ie.satisfaction) and probability of getting this utility.

SAMPLE OF PLAY FOR CALIS

III. RECREATIONAL AREA

```
 IN WHICH LIFE AREA WOULD YOU LIKE TO MAKE DECISIONS?
? 2


 NOTICE

 YOUR EXPECTED OVERALL SCHOOL GRADE IS A   FOR THIS YEAR.

 WHAT WOULD YOU LIKE TO DO?
? 1
```

Note: this is the delayed feedback message from the education area.

```
 WHAT SPORT WOULD YOU LIKE TO PARTICIPATE IN ?
 PLEASE TYPE IN YOUR OWN CHOICE.
? SWIMMING
 HOW MANY HOURS/WK WOULD YOU LIKE TO SPEND ON THIS ACTIVITY ?
? 3
 DO YOU OWN THE NECESSARY EQUIPMENT? 1=YES 0=NO
? 1
 DO YOU WANT TO CHOOSE ANOTHER RECREATIONAL ACTIVITY     THIS YEAR? 1=YES 0=NO
? 1
 WHAT WOULD YOU LIKE TO DO?
? 3  .

 HOW MANY HOURS/WK WOULD YOU LIKE TO SPEND ON THIS ACTIVITY ?
? 2
 WHAT WILL YOU DO TOGETHER ?
 PLEASE TYPE IN YOUR OWN  ANSWER.
? GO ON A PICNIC
 DO YOU WANT TO CHOOSE ANOTHER RECREATIONAL ACTIVITY     THIS YEAR? 1=YES 0=NO
? 1
 WHAT WOULD YOU LIKE TO DO?
? 10

 WHERE WOULD YOU LIKE TO GO ON VACATION ?
 PLEASE TYPE IN YOUR OWN ANSWER.
? EUROPE
 HOW MANY WEEKS WOULD YOU LIKE TO SPEND ON VACATION?
? 2
 HOW MUCH MONEY DO YOU WANT TO SPEND ON THIS VACATION?
? 500
 DO YOU WANT TO CHOOSE ANOTHER RECREATIONAL ACTIVITY     THIS YEAR? 1=YES 0=NO
? 1
 WHAT WOULD YOU LIKE TO DO?
? 11

 WHERE WOULD YOU LIKE TO GO WITH YOUR DATE ? (ASSUMING YOCAN
 GET ONE.---PLEASE TYPE IN YOUR OWN CHOICE.
? MOVIE
 HOW MANY HOURS/WK WOULD YOU LIKE TO SPEND ON THIS ACTIVITY ?
? 3
 HOW MUCH MONEY DO YOU WANT TO SPEND /WK ON DATES ?
? 7
 DO YOU WANT TO CHOOSE ANOTHER RECREATIONAL ACTIVITY     THIS YEAR? 1=YES 0=NO
? ???
 YOU PRESENTLY HAVE THE FOLLOWING RESOURCES REMAINING THIS YEAR
 NUMBER OF HOURS =            860 (    16 /WEEK)
 NUMBER OF DOLLARS = $    2086.00 (    40.12 /WEEK)
 AMOUNT IN SAVINGS ACCOUNT = $       0.00
```

Input provides for player to determine own activities

Example of On-demand Resource Check

Decision Options

1= Sports Activities
2= Visit Friends
3= Visit Relatives
4= Watch Television
5= Read a Book
6= Hobbies or Crafts
7= Listen to Music
8= Club,Union activities
9= Church Activities
10= Vacation
11= Dating Activities
12= Leave this area

SAMPLE OF PLAY FOR CALIS

IV. OCCUPATIONAL AND JOB DECISION AREA

```
 3
  ~
 WOULD YOU LIKE TO APPLY FOR A
 1=PART TIME JOB
 2=AFULL TIME JOB ?
 NOTE  YOU MAY HOLD BOTH JOB TYPES AT THE SAME TME
? 1
  ~
  ~
 IN WHICH AREA WOULD YOU LIKE TO GET A JOB ?
 1=CRAFTSMAN(EG.BAKER,BRICKLAYER,MECHANIC,ELECTRICIAN,TAILOR)
 2=LABORER (EG.GROUNDSKEEPER,CONSTRUCTION WORK)
 3=EQUIPTMENT OPERATOR(EG.ASSEMBLER,MEAT CUTTER,FURNACEMAN)
 4=CLERICAL(EG.BOOKEEPER,SECRETARY    ,TELEPHONE OPERATOR)
 5=TRANSPORTATION(EG.TRUCK DRIVER,TAXI DRIVER,BUS DRIVER)
 6=SALES WORK(EG.SALESPERSON,INSURANCE AGENT,ADVERTISING AGENT)
 7=SERVICE WORK(EG.JANITOR,HAIRDRESSOR,POLICE OFFICER,WAITER)
 NOTE  ALL POSITIONS ARE JUNIOR GRADE POSITIONS.
? 3
  ~
```

← Job Classifications from the major areas of the 1970 U.S. Census

```
 YOU MAY APPLY FOR ONE OF THE FOLLOWING POSITIONS
    1=FACTORY WORKER-PARTS ASSEMBLER-SEWER,STITCHER
    2=INSPECTOR(EG.MANUFACTORED GOODS,PRODUCE)
    3=FURNACEWORKER (METALSMELTER)
    4=GAS STATION,GARAGE ATTENDENT
    5=LAUNDRY OR DRYCLEANER
    6=MEAT CUTTER
    7=COAL MINER
    8= PACKER
    9=PUNCH PRESS,LATHE,STAMP PRESS,DRILL PRESS OPERATOR
    10=TEXTILE MILL WORKER
    11= WELDER OR RIVETER
? 4
```

← Job Choices within the 'Equipment Operator' Occupation Area

```
 AT HOW MANY PLACES OF EMPLOYMENT WOULD YOU LIKE TO APPL Y?
? 10
 HOW SATISIFYING IS GETTING THIS JOB?
  --------------------
? -----------------------------------
 CHANCES OF GETTING SATISFACTION
  ------------------
? -----------------------------------
  ~
```

← Subjective Utility and Probability measures

```
 YOU ARE OFFERED PART-TIME POSITION AS A GAS STATION ATTENDENT
     AT $3.00/HR. WILL YOU ACCEPT THE POSITION ? 0=NO 1=YES
? 1
```

Note: All part time positions start at $3.00/hr

```
 HOW MANY HOURS PER WEEK DO YOU WANT TO WORK? LT. 40
? 20 .
 YOU PRESENTLY HAVE THE FOLLOWING RESOURCES REMAINING THIS YEAR
 NUMBER OF HOURS =            1500 (    28 /WEEK)
 NUMBER OF DOLLARS = $   2950.00 (    56.73 /WEEK)
 AMOUNT IN SAVINGS ACCOUNT = $       0.00
 AMOUNT IN DEBT     = $       0.00
?
```

← Automatic Resource Check

SAMPLE OF PLAY FOR CALIS

V. FINANCES DECISION AREA

```
   ~
 IN WHICH LIFE AREA WOULD YOU LIKE TO MAKE DECISIONS?
? 5
   ~
 WOULD YOU LIKE TO
 1=BUY SOMETHING
 2=SELL SOME   THING
 3=SAVE SOME MONEY
 4=TAKE OUT A LOAN
? 3
   ~
   ~
 YOU HAVE $       0.00 IN YOUR SAVINGS ACCOUNT.
 DO YOU WANT TO
 1=DEPOSIT OR 2=WITHDRAW MONEY?
? 1  .
 HOW MUCH MONEY DO YOU WISH TO TRANSFER?
? 1000
 DO YOU WISH TO SPEND MORE TIME IN THIS AREA?
? 1
 WOULD YOU LIKE TO
 1=BUY SOMETHING
 2=SELL SOME   THING
 3=SAVE SOME MONEY
 4=TAKE OUT A LOAN
? 1
   ~
 WHAT WOULD YOU LIKE TO DO?
? 3  .
 HOW MANY WEEKS DO YOU WANT TO SPEND SHOPPING?
? 1
 HOW MUCH TIME/WK DO YOU WANT TO SPEND ON THIS ACTIVITY?  (INCLUDING SHOPPING)
? 3
   ~
   ~
 IS THIS YOUR FIRST AUTO BUY? 1=YES 0=NO
? 1
 DO YOU WANT TO BUY  1=NEW OR 0=USED CAR
? 0
 WHAT MAKE OF CAR ARE YOU INTERESTED IN?
 PLEASE       TYPE IN YOUR OWN CHOICE.
? OLD VOLKSWAGEN
 HOW MUCH DO YOU WANT TO SPEND ON THIS CAR?
? 500
 DO YOU WISH TO SPEND MORE TIME IN THIS AREA?
? 0 .
   ~
 IN TERMS OF ALL DECISIONS MADE IN THIS AREA, HOW
   SATISFIED ARE YOU?

?
```

Decision Options

1= Clothing
2= Sports Equipment
3= Automobile
4= Boat
5= Furniture
6= Home improvements
7= Books/Magazines
8= Television
9= Hobby/Craft Equipment
10= Records/Tapes
11= Purchase a Home
12= Rent Apartment
13= Alcohol and Cigarettes
14= Leave Area

Bibliography

Boocock, S. (1968) "An experimental Study of the Learning effects of Two Games with Simulated Environments" in Boocock, S. and E. O. Schild (eds.) Simulation Games in Learning. Beverly Hills, Calif., Sage Publications.

Boocock, S. (1972) "Validity Testing of an Intergenerational Relations Game." Simulation and Games 3 (March) 29-40.

Coleman, J. (1969) "Games as Vehicles for Social Theory" American Behavioral Scientist 12 (July/Aug):2-6.

Hermann, C. F. (1967) "Validation problems in Games and Simulations with special reference to international politics." Behavioral Science 12:216-231.

Shubki, M. (1970) "Teaching Economics with Competitive Games." Paper presented at the Ninth Symposium of the National Gaming Council, Washington, D.C.

Tallman, I. (1973) "Social Structure and Socialization for Change" University of Minnesota (mimeo)

Tallman, I. and L. Wilson (1975) "SIMCAR: A game simulation method for Cross National Family Research" Social Science Information.

Tallman I. and L. Wilson (1974) "Simulating Social Structure: The Use of a simulation Game in Cross National Research" Simulation and Games V,2 (June):147-167.

FOUR DIMENSIONAL BAR PLOTTING AS AN ALTERNATIVE DATA ANALYTIC METHODOLOGY

Marie E. Childers
Graduate School
Indiana University

David Lambert
Wrubel Computing Center
Indiana University

One of the major weaknesses of traditional statistical research techniques lies in the area of analyses of data that are "nominal" or "categorical" in nature. With data of this type researchers generally must opt for a class of statistical techniques called "non-parametric" statistics. Non-parametric statistical techniques make very few assumptions about the characteristics of the distributions of population from which the data are drawn. In comparison with parametric statistical techniques which are appropriate for data of ratio or interval type non-parametric statistics are less powerful and more conservative.

Without realizing the existence of these types of constraints novice researchers very often spend countless hours collecting data that are categorical in nature to explore hypotheses that involve establishing complex relationships among sets of nominally scaled variables. Upon entering the analysis phase of their research they then discover that they are unable to do tests of their hypotheses, to examine patterns of interactions among variables, and to deal adequately with the repeated measures aspects of their data sets. The path out of this dilemma then becomes a major impediment to the research process and often results in either catapaulting the researchers to a mode of analysis that is beyond their capabilities or causing decisions to be made that turn the final product into a maze of tables presenting relationships that are trivially simple (and normally bivariate in nature) or so complex as to be beyond interpretation by mere mortals.

The researcher who chooses the first path would most likely opt for techniques that allow for the statistical analyses of complex contingency tables. Such techniques however are relatively new additions to the statistical literature and involve complex concepts of long-linear modeling. While several computer programs exist for this type of analysis (here at Indiana University versions of NONMET, GENCAT, ECTA, and MULQUAL are available of use) most are in primitive states and not easily useable by novice researchers. Nor are they particularly suitable for use in an exploratory research mode, since nearly all are geared toward testing specific models.

The second path is probably more accessable to the novice researcher who is quite capable of generating SPSS CROSSTABS tables and reporting non-parametric statistics. Even though these statistics are rendered inappropriate due to large scale violations of assumptions for the tests.

Fortunately researchers at Indiana University have a more appealing path out of this research dilemma. FOUBAR, a four dimensional histogram plotting package interfaced with SPSS developed at Wrubel Computing Center by Jim Hettmer and Mickey Stentz is an ideal vehicle for a researcher to examine complex relationships among categorical variables. The capability to visually demonstrate relationships not only establishes FOUBAR as a tool for data analysis but elevates it to the status of a research methodology. Exploratory analysis is enhanced by the capacity to start out with simple univariate presentations of data and proceed to add on additional levels as one proceeds to explore the more complex relationships in the data set. The following several pages present a summary of a case study in using FOUBAR as an alternative data analytic methodology.

The literature applying organization theory to the functioning of institutions of higher education is limited. Empirical research on the business sector has richer, as well as more lengthy history. Studies in both higher education and business, however, have been done on three main types of decision-making structures, (1) bureaucratic structure, (2) collegial structure, and (3) political structure. The focus of this research was on the three forms of decision making but unlike other studies, it compared the extent to which each of the structures was used on university campuses.

The problem of the study was to determine: whether multiple decision-making structures were utilized in university governance, whether particular decision-making forms were used for specific categories of decisions, whether perceptions of decision-making structures varied according to category of institution or by status of the rater, and whether perceived decision-making structures were the same as the preferred structure.

The conceptual framework of the research comprised eight dimensions which cut across all three decision-making forms: basis of the decision maker's power, implementation of the decision, most likely evaluation criterion, who approved the decision, decision maker's autonomy, decision resolution, control utilized by the decision maker and mode of conflict resolution. A summary of the distinctions among dimensions can be found in Chart 1.

The conceptual framework provided the basis for developing an instrument used to investigate responses about fifteen academic decisions. The fifteen decision, divided into three decision categories can be found in Chart 2. Respondents were also asked to identify the individual or group, by category, that was the most responsible decision maker for the given decision. For this purpose, there were two administrator categories and three faculty categories.

The design was a quasi-experimental field design. The fifteen decisions were independent variables. The modering variables were the decision-making dimensions which formed the bureaucratic, collegial, and political structures. Dependent variables were what structure existed (perceived), what structure should prevail (preferred), and the correspondence between the two or the degree of satisfaction. The instrument was personally administered to 54 faculty members and 54 administrators in six universities. The universities were matched by types of control (public or independent) and by a general classification system related to the emphasis of their mission. The institution categories were public comprehensive universities, public research universities and private universities.

Chart 1

Distinctions Among Decision-Making Structures

	Bureaucratic	Collegial	Political
Bases of Power	Position	Academic or Professional Expertise	Capacity to Reward or Punish
Implementation of Decision	Leadership Skill	Academic or Professional Ability	Political Skill
Criterion Used to Evaluate Decision	Smooth Functioning of of the University	Educational Merit	Protection of Political Interests
Approval of the Decision	Administrative Review	Faculty Review	Political Review
Autonomy of the Decision Maker	Within Prescribed Guidelines	Within Academic or Professional Competency	Within Political Constraints
Resolution to the Decision	Within the Bureaucratic Structure	Within the Collegial Structure	Within the Political Structure
Control by Decision Maker	Budget Control	Personal Persuasion and Credibility	Use of Power Relationships
Conflict Resolution	Through Authoritarian Decisions	Through Collegial Consensus	Through Bargaining and Negotiation

Chart 2

Academic Decisions by Category

Faculty Personnel Decisions	Institutional Management Decisions	Academic Mission Decisions
Determining Merit Raises for Faculty	Allocating Financial Resources for College or School Budgets	Initiating Grant Proposals for the Funding of Academic Activities
Granting Faculty Leaves and Sabbaticals	Determining Space and Facility Allocations	Modifying Degree Requirements
Recruiting a Pool of Faculty Candidates	Scheduling Instructional Activities	Deciding Who Is to Be Admitted to a Graduate Program
Selecting Faculty for Appointment	Authorizing the Use of University Resources by the Community	Revising Course Requirements or Structure (Departmental)
Evaluating Faculty for Promotion or Tenure	Resolving Unusual or Emergency Situations	Developing New Course Material (Adding a new course to the curriculum)

The exploratory nature of the research allowed many interesting relationships to emerge. One major finding was that, in general, all academic mission decisions were perceived to be collegial in form. Figure 1 and Table 1 displays this finding. According to the conceptual framework of the study (Chart 1), the collegial structure meant the following:

1. The individual or group having major responsibility for the decision has power based on academic or professional expertise.
2. The decision is implemented through the academic or professional ability of the decision maker.
3. The decision is most likely to be evaluated in terms of its educational merit.
4. The primary approval of the decision is usually through faculty review.
5. The individual or group having the major decision-making responsibility has autonomy of action within academic or professional competencies.
6. A decision is resolved or reaches its conclusion within the collegial structure.
7. The individual or group making the decision conctrols or directs action in keeping with the spirit of the decision through personal persuasion.
8. If conflict over the decision arises, it is resolved through collegial consensus.

A complementary finding was that institutional management decisions had mostly bureaucratic structural characteristics. Figure 2 and Table 2 presents the finding. In this study, a bureaucratic structure means the following:

1. The individual or group having major responsibilty for making the decision has power because of the position(s) occupied.
2. A decision is implemented through the leadership skill of the individual or group.
3. A decision is primarily evaluated on some criterion that relates to the smooth functioning of a university.
4. The primary approval of a decision is usually through administrative review.
5. The individual or group making the decision has autonomy of action within institutionally prescribed guidelines.
6. A decision is resolved or concluded within the bureaucratic structure.

Figure 1

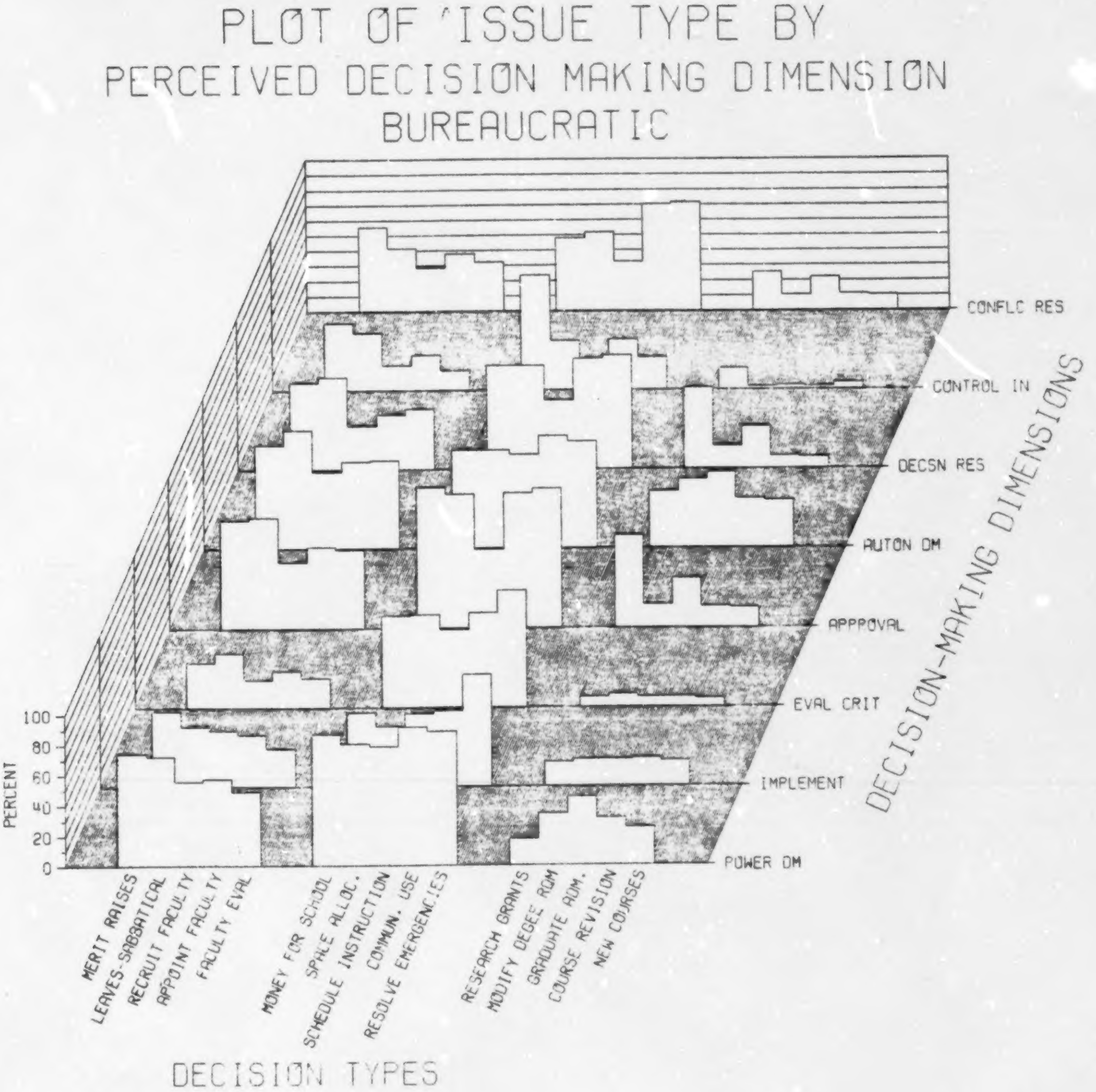

PLOT OF 'ISSUE TYPE BY
PERCEIVED DECISION MAKING DIMENSION
BUREAUCRATIC
PERCENT
0
20
40
60
80
100
CONFLC RES
CONTROL IN
DECSN RES
AUTON DM
APPROVAL
EVAL CRIT
IMPLEMENT
POWER DM
DECISION-MAKING DIMENSIONS
MERIT RAISES
LEAVES-SABBATICAL
RECRUIT FACULTY
APPOINT FACULTY
FACULTY EVAL
MONEY FOR SCHOOL
SPACE ALLOC.
SCHEDULE INSTRUCTION
COMMUN. USE
RESOLVE EMERGENCIES
RESEARCH GRANTS
MODIFY DEGEE RQM
GRADUATE ADM.
COURSE REVISION
NEW COURSES
DECISION TYPES

TABLE 1

TABLE OF ISSUE TYPE BY PERCEIVED DECISION MAKING DIMENSION BUREAUCRATIC

DECISION-MAKING DIMENSIONS	DECISION TYPES														
	MERIT RAISES	LEAVES-SABBATIC	RECRUIT FACULTY	APPOINT FACULTY	FACULTY EVAL	MONEY FOR SCHOOL	SPACE ALLOC.	SCHEDULE INSTRUC	COMMUN. USE	RESOLVE EMERGENC	RESEARCH GRANTS	MODIFY DEGEE RQM	GRADUATE ADM.	COURSE REVISION	NEW COURSES
POWER DM	74.1	72.2	55.6	57.4	49.1	87.0	80.6	78.7	91.7	88.9	17.8	34.3	45.4	31.5	25.0
IMPLEMENT	50.0	39.8	37.0	34.3	25.0	49.1	39.8	48.1	50.0	74.1	15.9	17.6	17.6	18.5	16.7
EVAL CRIT	29.6	36.1	18.5	24.1	19.4	60.4	61.1	50.9	62.0	76.9	6.5	8.3	6.5	6.5	5.6
APPROVAL	71.3	73.1	43.5	53.7	51.9	92.6	88.0	51.9	88.9	91.7	60.7	14.8	32.4	13.9	13.0
AUTON DM	68.5	78.7	50.9	57.4	58.3	64.8	64.8	62.0	74.1	70.4	37.4	45.4	50.0	32.4	31.5
DECSN RES	57.4	61.1	28.7	36.1	39.8	68.5	68.5	45.4	73.1	75.0	53.3	14.8	27.8	8.3	7.4
CONTROL IN	44.4	38.0	16.7	23.1	13.0	75.9	32.4	17.6	33.3	21.3	14.0	1.9	2.8	1.9	4.6
CONFLC RES	55.6	41.7	28.7	38.0	32.4	48.1	51.9	32.4	70.4	71.3	25.2	10.2	22.2	11.1	10.2

Figure 2

PLOT OF ISSUE TYPE BY
PERCEIVED DECISION MAKING DIMENSION
COLLEGIAL

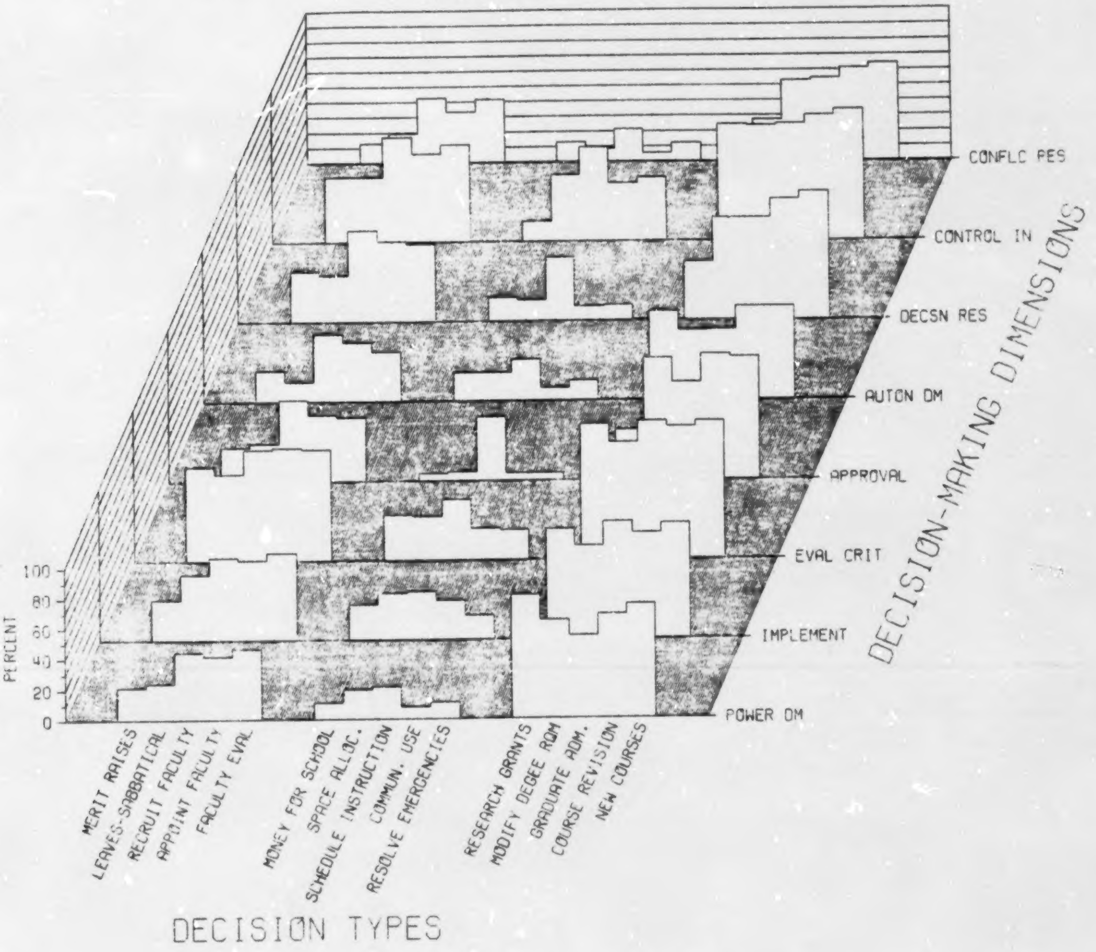


TABLE 2

TABLE OF ISSUE TYPE BY PERCEIVED DECISION MAKING DIMENSION COLLEGIAL

DECISION-MAKING DIMENSIONS	DECISION TYPES														
	MERIT RAISES	LEAVES-SABBATIC	RECRUIT FACULTY	APPOINT FACULTY	FACULTY EVAL	MONEY FOR SCHOOL	SPACE ALLOC.	SCHEDULE INSTRUC	COMMUN. USE	RESOLVE EMERGENC	RESEARCH GRANTS	MODIFY DEGEE RQM	GRADUATE ADM.	COURSE REVISION	NEW COURSES
POWER DM	21.3	24.1	44.4	41.7	46.3	11.1	19.4	21.3	8.3	11.1	81.3	64.8	54.6	68.5	75.0
IMPLEMENT	26.9	43.5	54.6	52.8	57.4	23.1	30.6	31.5	25.9	15.7	72.0	61.1	76.9	69.4	75.9
EVAL CRIT	62.0	57.4	73.1	74.1	73.1	29.6	28.7	39.8	20.4	19.4	88.8	76.9	90.7	87.0	90.7
APPROVAL	22.2	25.0	53.7	43.5	41.7	4.6	5.6	41.7	4.6	4.6	32.7	79.6	63.9	82.4	80.6
AUTON DM	20.4	13.0	44.4	38.9	32.4	18.5	18.5	26.9	8.3	13.0	57.9	45.4	46.3	61.1	61.1
DECSN RES	33.3	30.6	60.2	52.8	50.9	15.7	13.9	41.7	9.3	10.2	38.3	67.6	67.6	79.6	84.3
CONTROL IN	42.6	42.6	69.4	58.3	63.9	13.0	43.5	62.0	38.0	41.7	76.6	75.9	76.9	82.4	86.1
CONFLC RES	13.0	19.4	42.6	33.3	41.7	13.0	8.3	21.3	5.6	12.0	27.1	52.8	54.6	61.1	63.9

7. A person or group making the decision controls or directs action in keeping with the intent of the decision through budget control.

8. Conflict, if it occurs, is resolved through authoritarian decisions.

In the faculty personnel decisions, three out of five were perceived to have substantial but not complete collegial structure. The three decisions were recruiting a pool of faculty candidates, selecting faculty for appointment and evaluating faculty for promotion and tenure. The other two decisions, determining merit raises for faculty members and granting faculty leaves and sabbaticals, were seen to have many but not all of the bureaucratic structural characteristics. Figures and Tables 1 and 2 display the finding.

Figures 3, 4, 5, present another way of displaying the same data, only this time the data reported by institution category.

The most responsible decision makers for each of the decisions are displayed in Figure 6 and Table 3.

The research treated the bureaucratic, collegial, and political decision-making forms as independent structures, an assumption made in most of the literature. The results of the factor analysis challenged the orthogonal nature of the structures and suggested that the bureaucratic and collegial forms in particular were probably part of the same structure -- each on either end of a continuum.

The previous bureaucratic and collegial findings were analyzed by specific decision and dimension, both of which contributed, in general, to meaningful distinctions among the data. Institutional category also was a classification which seemed to produce distinct results. In the perceived and preferred results, but not the satisfaction results, the classification, faculty vs. administrator, did not lead to meaningful differences. The results were displayed in eighty-four tables and graphs, the volume of which prohibits an abbreviated summary of results.

The following hypotheses were confirmed as strong relationships in the data.

1. Multiple decision-making forms operate on any given campus.

2. Institutional management decisions are usually bureaucratic in decision-making structure.

3. Academic mission decisions are usually collegial in structure.

4. Faculty personnel decisions are both bureaucratic and collegial in decision-making structure.

5. There is limited support for the notion that universities emphasizing research indicate the most collegiality for academic mission decisions. Academic mission decisions are only slightly more collegial in public research universities.

Figure 3

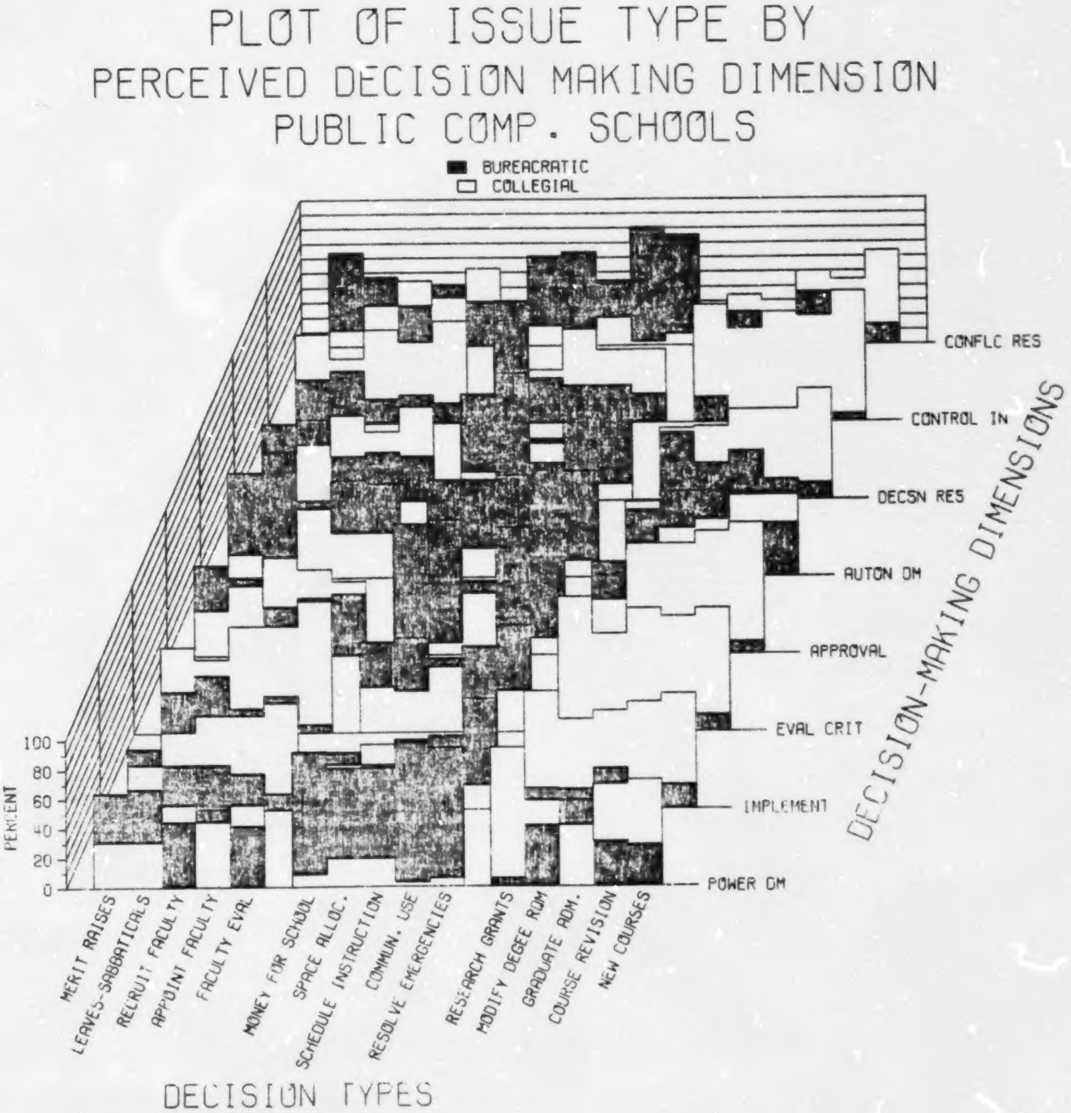

PLOT OF ISSUE TYPE BY
PERCEIVED DECISION MAKING DIMENSION
PUBLIC COMP. SCHOOLS
BUREACRATIC
COLLEGIAL
PERCENT
0
20
40
60
80
100
CONFLC RES
CONTROL IN
DECSN RES
AUTON DM
APPROVAL
EVAL CRIT
IMPLEMENT
POWER DM
DECISION-MAKING DIMENSIONS
MERIT RAISES
LEAVES-SABBATICALS
RECRUIT FACULTY
APPOINT FACULTY
FACULTY EVAL
MONEY FOR SCHOOL
SPACE ALLOC.
SCHEDULE INSTRUCTION
COMMUN. USE
RESOLVE EMERGENCIES
RESEARCH GRANTS
MODIFY DEGEE RQM
GRADUATE ADM.
COURSE REVISION
NEW COURSES
DECISION TYPES

Figure 4

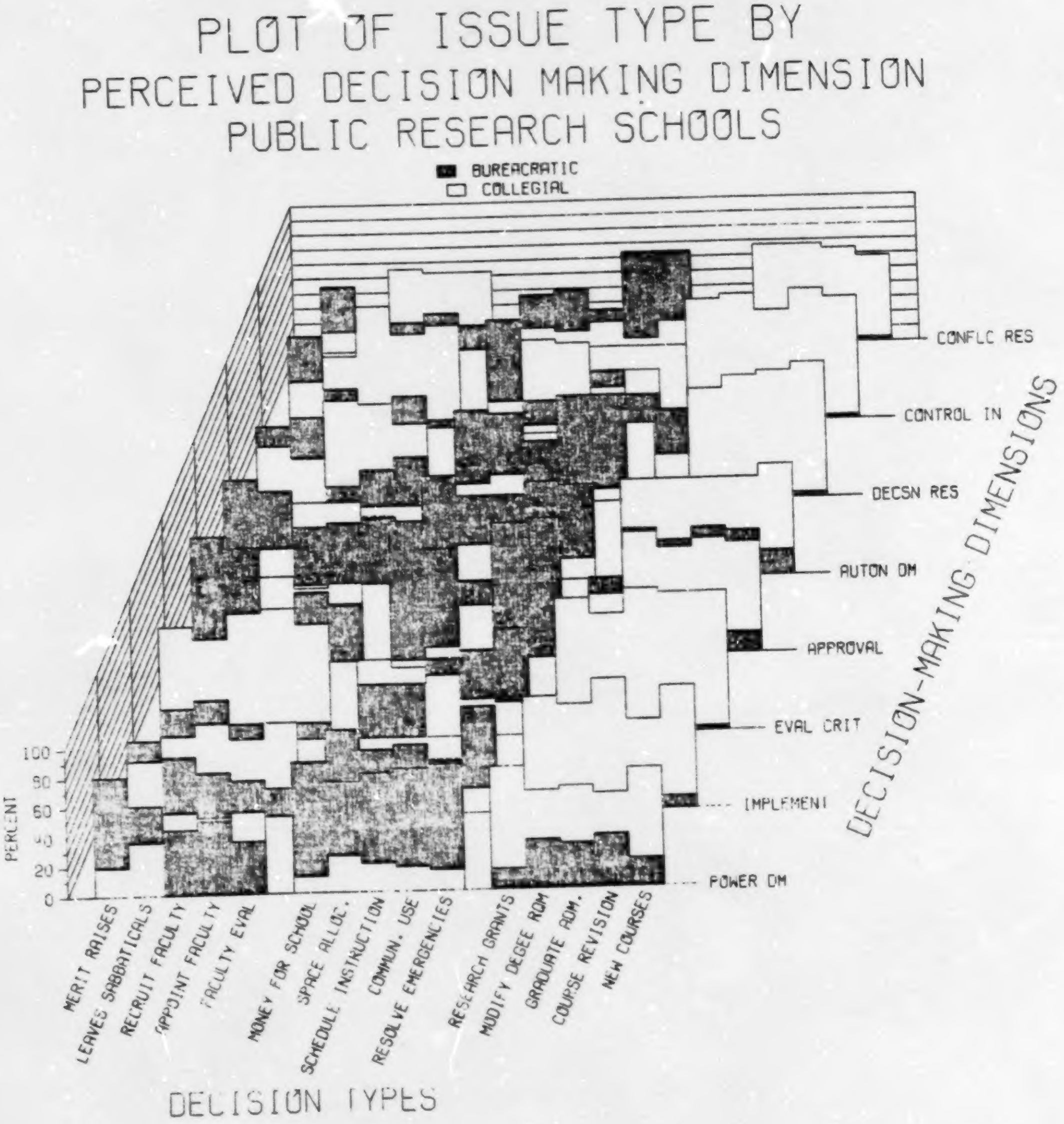
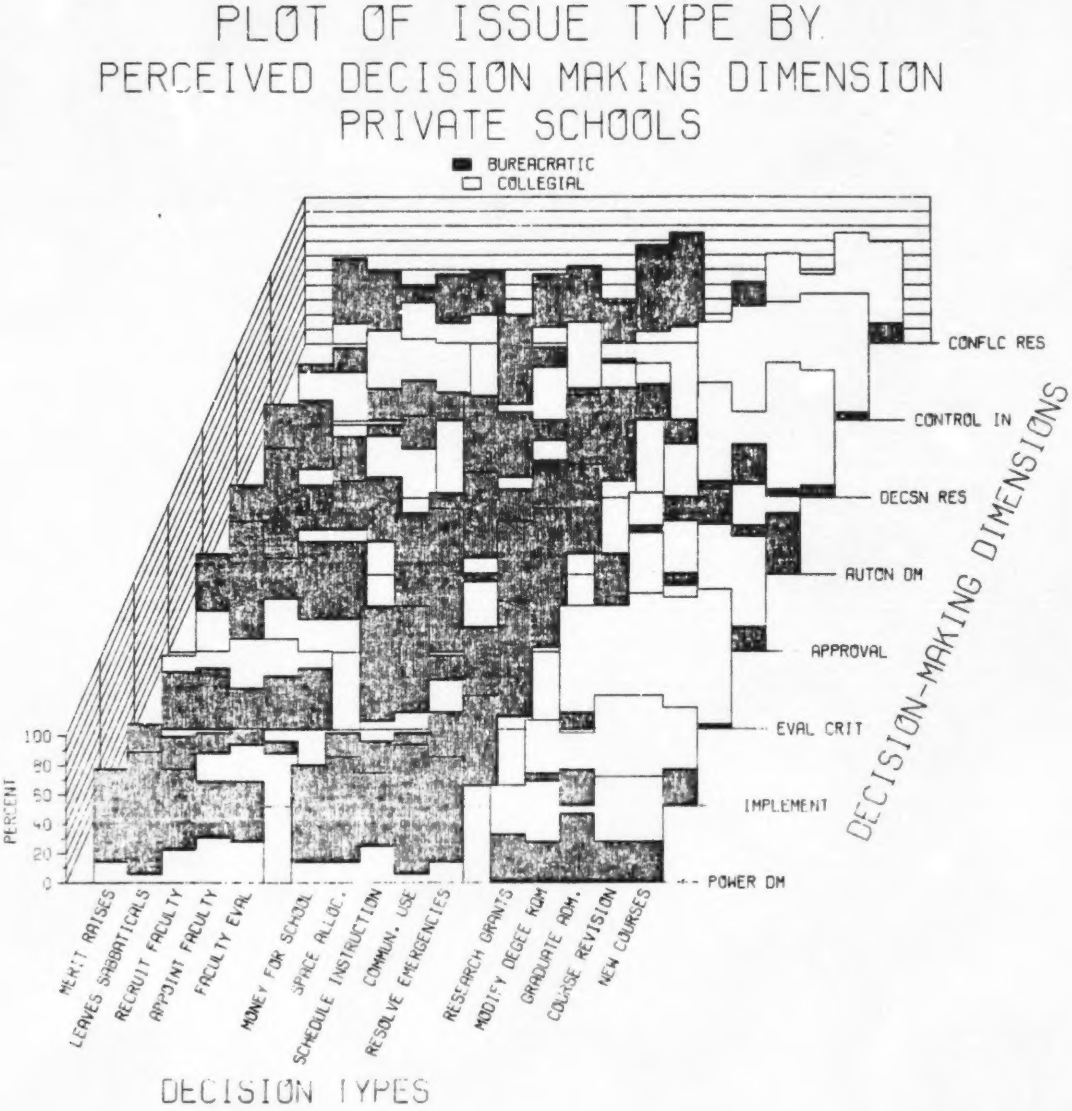

PLOT OF ISSUE TYPE BY
PERCEIVED DECISION MAKING DIMENSION
PUBLIC RESEARCH SCHOOLS
▨ BUREACRATIC
☐ COLLEGIAL
PERCENT
0
20
40
60
80
100
MERIT RAISES
LEAVES SABBATICALS
RECRUIT FACULTY
APPOINT FACULTY
FACULTY EVAL
MONEY FOR SCHOOL
SPACE ALLOC.
SCHEDULE INSTRUCTION
COMMUN. USE
RESOLVE EMERGENCIES
RESEARCH GRANTS
MODIFY DEGEE RQM
GRADUATE ADM.
COURSE REVISION
NEW COURSES
DECISION TYPES
CONFLC RES
CONTROL IN
DECSN RES
AUTON DM
APPROVAL
EVAL CRIT
IMPLEMENT
POWER DM
DECISION-MAKING DIMENSIONS

Figure 5



HOLSWVSA END OF PLOT.

Figure 6

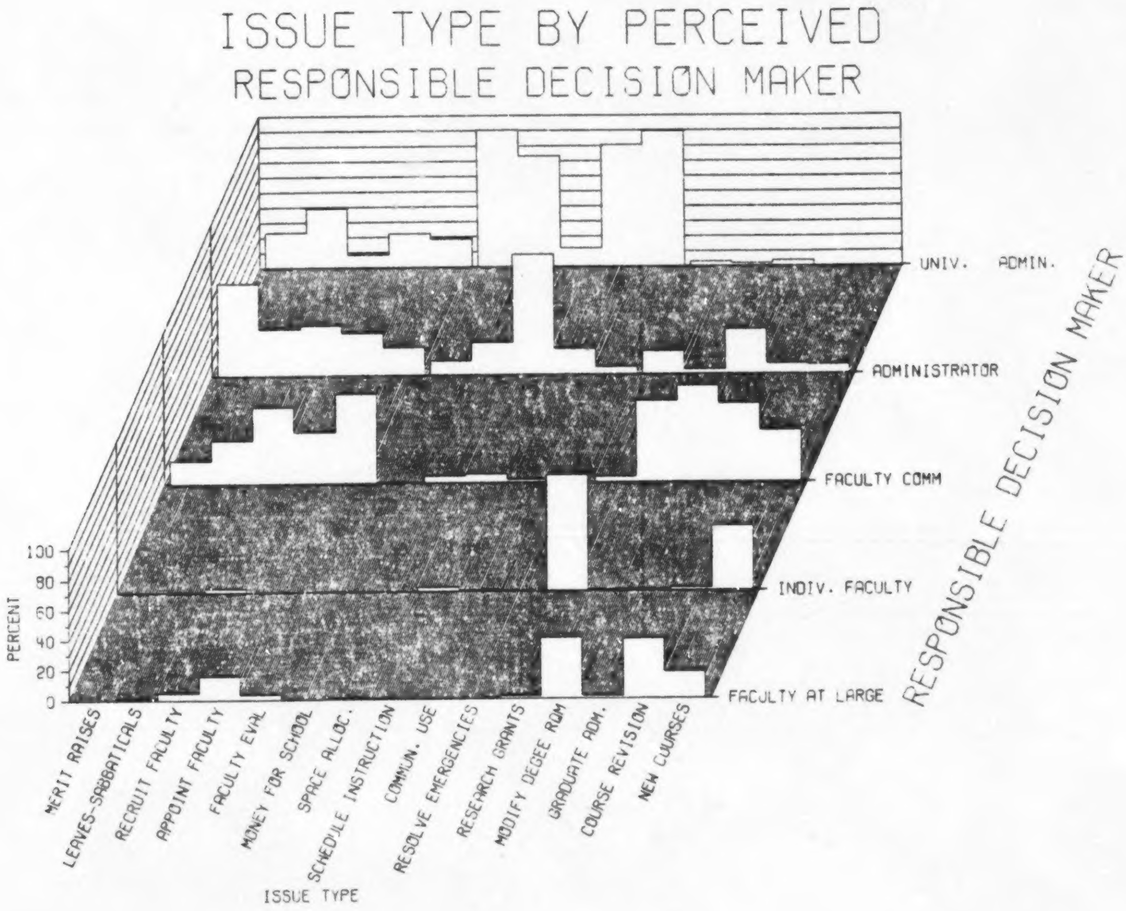

ISSUE TYPE BY PERCEIVED
RESPONSIBLE DECISION MAKER
PERCENT
100
80
60
40
20
0
UNIV. ADMIN.
ADMINISTRATOR
FACULTY COMM
INDIV. FACULTY
FACULTY AT LARGE
RESPONSIBLE DECISION MAKER
MERIT RAISES
LEAVES-SABBATICALS
RECRUIT FACULTY
APPOINT FACULTY
FACULTY EVAL
MONEY FOR SCHOOL
SPACE ALLOC.
SCHEDULE INSTRUCTION
COMMUN. USE
RESOLVE EMERGENCIES
RESEARCH GRANTS
MODIFY DEGEE RQM
GRADUATE ADM.
COURSE REVISION
NEW COURSES
ISSUE TYPE

TABLE 3

ISSUE TYPE BY PERCEIVED RESPONSIBLE DECISION MAKER

RESPONSIBLE DECISION MAKER	ISSUE TYPES														
	MERIT RAISES	LEAVES-SABBATICALS	RECRUIT FACULTY	APPOINT FACULTY	FACULTY EVAL	MONEY FOR SCHOOL	SPACE ALLOC.	SCHEDULE INSTRUCTION	COMMUN. USE	RESOLVE EMERGENCIES	RESEARCH GRANTS	MODIFY DEGEE RQM	GRADUATE ADM.	COURSE REVISION	NEW COURSES
FACULTY AT LARGE	0.0	.9	4.6	15.7	3.7	0.0	.9	0.0	0.0	0.0	1.9	40.7	1.9	39.8	17.9
INDIV. FACULTY	0.0	0.0	2.8	0.0	0.0	0.0	0.0	2.8	.9	0.0	76.6	.9	.9	1.9	42.5
FACULTY COMM	15.7	29.0	50.9	34.3	59.3	0.0	4.6	5.6	1.9	5.6	3.7	53.7	63.9	51.9	34.0
ADMINISTRATOR	61.1	30.8	33.3	27.8	18.5	9.3	21.3	79.6	16.7	4.6	15.0	2.8	29.6	6.5	5.7
UNIV. ADMIN.	23.1	39.3	8.3	22.2	18.5	90.7	73.1	12.0	80.6	89.8	2.8	1.9	3.7	0.0	0.0

6. Satisfaction levels are the same in all institution categories even though public research universities exhibit slightly higher satisfaction.

The following hypothesis was not confirmed as a strong relationship.

1. When perceived bureaucratic decision making exists for faculty personnel decisions, the preferred decision-making pattern will be collegial in form. Although the hypothesis holds in general, the exceptions would suggest that the perceived-preferred shift in decision-making structure is dependent upon the institution setting.

In summary, the exploratory research was successful in developing an instrument which tested the presence of bureaucratic, collegial, and political decision making in diverse university settings. The presentation of the data through graphs or figures was an effective method of grasping the multi-dimensionality of the phenomena in the study. It was a method whereby the complex nature of university decision making could be communicated to people with varying degrees of sophistication.

HYPAT: A Program to Produce Hyperbolic Patterns

Howard Alexander
Professor of Mathematics Emeritus
Earlham College

The program HYPAT, described in this paper, is intended to produce a type of repeating design. These designs resemble the patterns we find in fabrics, wallpaper and floor coverings, in this respect: they consist of many repetitions of a basic motif. The motif is a freely chosen shape: a leaf, a flower, a geometric shape or a "doodle". This shape is repeated by subjecting it to reflections and rotations in some systematic way. The commonly encountered designs (fabrics, etc.) are based on Euclidean geometry, whereas those we are here considering are based on non-Euclidean geometry; more specifically, on hyperbolic geometry. It is first necessary to explain what is meant by these terms.

Non-Euclidean geometry (invented in the 1820's) is a system of geometry very much like Euclid's, except that the idea of parallelism is left out. The result is that, in this system, the sum of the three angles of a triangle is no longer 180°. There are, in fact, two kinds of non-Euclidean geometry, depending on whether the angle sum of a triangle is greater than 180° (elliptic geometry) or less than 180° (hyperbolic geometry). Here we deal only with the second type. In fact, the name of the program, HYPAT, is an abbreviation for hyperbolic pattern.

To understand how patterns may be produced in hyperbolic geometry we first conceive the whole plane to be covered with regular polygons. Assume that each polygon has NA sides, and that NB of these polygons meet at each vertex. Such a covering of the plane is called an (NA,NB) tiling, and the regular polygons are called tiles. If we take a single tile and join its center to each of the vertices, and to the midpoints of the sides, the tile is divided into 2*NA right triangles. The three angles of this triangle will be (in degrees) 180/NA, 180/NB and 90. Since, in hyperbolic geometry the angle sum is less than 180°, we must have $180/NA + 180/NB + 90 < 180$, or

(1) $(NA - 2)*(NB - 2) > 4.$

Any solution of (1) yields an (NA,NB) tiling. Note that the "*" in (1) is used to indicate multiplication, in keeping with FORTRAN. There are infinitely many solutions of (1), of which a few are:

$$\begin{array}{cccc} (3,7), & (3,8), & (3,9), & \ldots \\ (4,5), & (4,6), & (4,7), & \ldots \\ (5,4), & (5,5), & (5,6), & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

If each of the tiles in an (NA,NB) tiling is divided into 2*NA triangles, as suggested earlier, and if the same motif is placed in each triangle, then we obtain a hyperbolic pattern. Our aim is to use the computer/plotter configuration to produce such designs.

It is not possible to represent hyperbolic geometry on the Euclidean plane without some distortion. However, there is an interesting model of the hyperbolic plane that was proposed in 1908 be Henri Poincaré. The whole hyperbolic plane is represented by the interior of a circle. The circle itself is called the absolute. The straight lines of the hyperbolic plane are represented by circular arcs which meet the absolute orthogonally, and by diameters of the absolute. The angle between two hyperbolic lines is simply the Euclidean angle between the circles (or lines) which represent the hyperbolic lines. In Figure 1, three hyperbolic lines are shown. Two are diameters of the absolute, meeting in the point A, which is the center of the absolute. Since the angles at B and C are each less than the corresponding angles of the Euclidean triangle whose third side, BC, is shown by means of a dotted line, it is evident that for this triangle the angle sum is less than 180°.

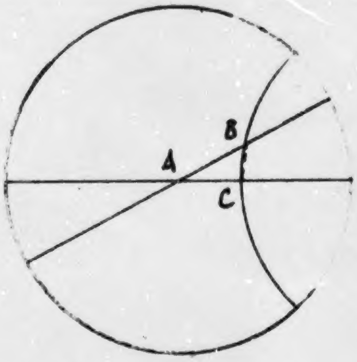


Figure 1

To express the Poincaré model in analytic terms we make use of a system of 3-vectors: $A = (a_1, a_2, a_3)$, $B = (b_1, b_2, b_3)$, ... and an inner product:

(2) $$A \cdot B = a_1b_1 + a_2b_2 - a_3b_3.$$

The absolute is taken to be a unit circle with its center at the origin of a system of Cartesian coordinates. A point P is represented by a vector $P = (p_1, p_2, p_3)$ with $P \cdot P = 1$. This corresponds to the Euclidean point (x_1, x_2) where

(3) $$x_1 = p_1/(1 + p_3), \quad x_2 = p_2/(1 + p_3).$$

Conversely, given the Euclidean point (x_1, x_2), the corresponding 3-vector is $P = (p_1, p_2, p_3)$ where

(4) $$p_1 = 2x_1/D, \quad p_2 = 2x_2/D, \quad p_3 = (2 - D)/D, \text{ where } D = 1-x_1^2-x_2^2.$$

A line A is represented by the vector $A = (a_1, a_2, a_3)$ with $A \cdot A = 1$. If $a_3 = 0$, this corresponds to the Euclidean line $a_1u_1 + a_2u_2 = 0$; if $a_3 \neq 0$, then A corresponds to the Euclidean circle with radius $1/|a_3|$ and center $(a_1/a_3, a_2/a_3)$. We are here using the same letter to refer to a vector and to the geometrical entity (point or line) to which it corresponds. With this understanding, we may say that the point P lies on the line A if and only if $A \cdot P = 0$. Finally, if A and B are two lines, and Θ is the angle between them, then

$$\cos \Theta = \pm(A \cdot B). \tag{5}$$

It turns out that, as soon as the parameters NA and NB have been selected, then the tiling is uniquely determined. Imagine that one of the tiles has been placed with its center at the origin and with one of its sides perpendicular to the x_1-axis. Of the 2*NA triangles into which this polygon may be subdivided, one will have one side coinciding with the x_1-axis and one side perpendicular to the x_1-axis, as shown in Figure 1. We call this the <u>initial triangle</u>. Given only the numbers NA and NB the coordinates of its vertices A, B and C can be computed. They turn out to be

A: (0, 0), B: (A3, 0), C: (A5, A6),

where the numbers A3, A5, A6 are computed as follows:

```
     A1 = COS(PI/NB)/SIN(PI/NA),     A2 = SQRT(A1*A1 - 1),
     A3 = A2/(1 + A1),   A4 = SQRT((A1*COS(PI/NA))**2 - A2*A2),
(6)
     A5 = A2*COS(PI/NA)/(A4 + COS(PI/NA)),
     A6 = A2*SIN(PI/NA)/(A4 + COS(PI/NA)).
```

The HYPAT program will print these coordinates as soon as the basic parameters are supplied. The user will then plot the initial triangle and draw a motif within it. Points are selected on the motif, and their coordinates are determined. When the basic parameters are entered a second time, together with the coordinates of the points, the design is executed. We first illustrate the whole procedure by means of an example, and then deal with the logic of the program construction.

<u>Example</u> Suppose we wish to create a (7, 3) design, that is, NA = 7, NB = 3. For the first run, two data cards are required: one with 1 in col. 1, which indicates that we are performing the first run; and one with 7 in col. 3 and 3 in col. 6, which gives the parameters NA and NB. The three vertices of the initial triangle will then be printed out:

(0, 0), (.2661, 0), (.2710, .1305) .

We plot these three points on a piece of graph paper, as shown in Figure 2, and join them to form the triangle ABC. Strictly speaking, the side BC should be the arc of a circle, as in Figure 1, rather than a straight line; but this will rarely make much difference. Within the triangle we now sketch a free-hand motif, which for our example is an abstract leaf-form. Thirty points have been located on the motif, and their coordinates are given in Table 1, in columns labeled X and Y. In the column labeled PEN we give the position of the plotter pen:

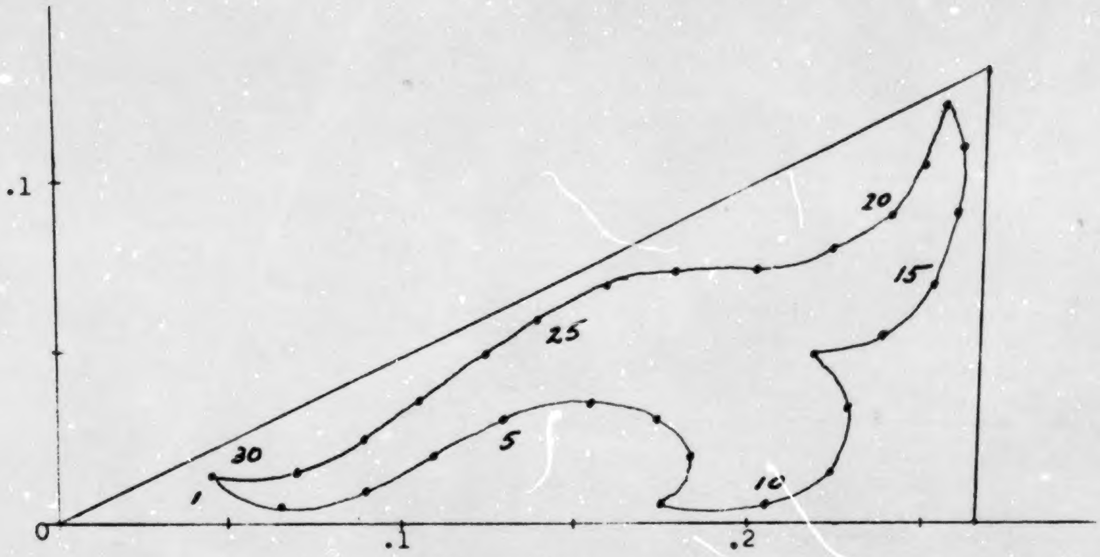


Figure 2

Table 1

#	X	Y	PEN	#	X	Y	PEN	#	X	Y	PEN
1	.045	.015	3	11	.225	.015	2	21	.227	.08	2
2	.065	.005	2	12	.23	.035	2	22	.205	.075	2
3	.09	.01	2	13	.22	.05	2	23	.18	.075	2
4	.11	.02	2	14	.24	.055	2	24	.16	.07	2
5	.13	.03	2	15	.255	.07	2	25	.14	.06	2
6	.155	.035	2	16	.262	.09	2	26	.125	.05	2
7	.175	.03	2	17	.263	.11	2	27	.105	.035	2
8	.185	.02	2	18	.26	.12	2	28	.09	.025	2
9	.175	.005	2	19	.253	.104	2	29	.07	.015	2
10	.205	.005	2	20	.244	.09	2	30	.045	.015	2

3 indicates that the pen is UP, 2 indicates that the pen is DOWN, as it moves from the last point to the current point. If the motif had consisted of several disjoint pieces, then the pen would be UP at the beginning of each piece. For the second run we need: first, a card with 2 in col. 1, to indicate that we are performing the second run; second, a card with 7 in col. 3, 3 in col. 6, 30 in cols. 8 and 9 (to give the number of points) and 8.0 in cols. 11-13 (to give the diameter of the limiting circle or absolute); finally, the 30 cards which give, in 3 fields of 4 cols. each, the X-value, the Y-value and the pen position.

The essential fact upon which the program HYPAT depends may be stated as follows: by repeated reflections in the three sides of the initial triangle the motif contained in that triangle may be repeated as often as we wish, thus covering a selected portion of the hyperbolic plane. To make this statement intelligible we need to do several things. First, we must define the term reflection. Then we must show how reflections may be carried out by means of matrix multiplication. Finally, we must show how the program HYPAT systematically repeats the motif over the chosen portion of the plane.

Since the straight lines of the hyperbolic plane are represented by Euclidean circles and straight lines, we need to define the idea of reflection in a quite general way. The idea we need is that of inversion in a straight line or circle. To define this concept, let C be any circle or straight line of the Euclidean plane, and let P be any point not lying on C. It may be shown that every circle or straight line which passes through P and is orthogonal to C also passes through a second point P' which is called the inverse of P in C (or with regard to C). In this definition nothing has been said about non-Euclidean geometry. We now define the reflection of a point P in a hyperbolic line A as the inverse of P in A.

We now proceed to use matrix multiplication as a means of expressing the operation of reflection analytically. To this end, let $A = (a_1, a_2, a_3)$ be any hyperbolic line, let P be any point, and let P' be its reflection in A. To prepare the way for matrix multiplication, we express P and P' as column vectors:

$$P = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}, \qquad P' = \begin{bmatrix} p'_1 \\ p'_2 \\ p'_3 \end{bmatrix}.$$

Let M_A denote the matrix

$$(7) \qquad M_A = \begin{bmatrix} 2a_1^2 - 1 & 2a_1a_2 & -2a_1a_3 \\ 2a_1a_2 & 2a_2^2 - 1 & -2a_2a_3 \\ 2a_1a_3 & 2a_2a_3 & -2a_3^2 - 1 \end{bmatrix}.$$

It can be shown that the reflection P' of P in A is given by

$$(8) \qquad P' = M_A P,$$

where matrix multiplication is to be understood on the right side of (8). Equation (8) has the consequence that if several reflections are performed in succession, then the matrix of the resultant operation is simply the product of the matrices of the several reflections. Because of the "essential fact" mentioned earlier it is sufficient to determine matrices for reflections in the three sides of the initial triangle ABC, which requires that we determine the vectors of the three sides. These turn out to be:

Side AC: (0, 1, 0) .

Side AB: $(\sin\Theta, -\cos\Theta, 0)$, where $\Theta = 180^\circ/\mathrm{NA}$, $\quad = 180^\circ/\mathrm{NB}$,
Side BC: $(m_1, 0, m_3)$, $\qquad m_1 = \cos\ /\sin\Theta$, $m_3 = \ /m_1{}^2 - 1$.

These vectors, together with (7), are used in HYPAT to construct matrices for the reflections in the three sides of the initial triangle.

HYPAT consists of a main program and three subroutnines EQ, PR and PAT. The subroutines EQ and PR are used to perform the operations of matrix equality and matrix multiplication, since we do not assume that the user will have access to a version of FORTRAN which provides these matrix operations. The array T(3,3,16) in COMMON consists of a "stack" of 16 3x3 matrices, which in this documentation we denote by $T_1, T_2, \ldots, T_{16}$. The subroutine EQ will make any one of these 16 matrices equal to any other, while the subroutine PR forms the product of any two of the matrices and stores the result in any other selected position in the stack. These operations will be illustrated presently.

To understand the structure of the program we need to consider again an (NA,NB) tiling. The main program selects a sequence of these tiles, and focuses on each tile in succession. Having selected a tile, the subrountine PAT is called to fill up that tile with repetitions of the chosen motif. It does this by placing a repetition of the motif in each of the 2*NA triangles into which the tile may be divided. Such a cluster of repetitions, filling up a given tile, we call a medallion. Thus the function of PAT is to create each of the medallions that make up the design.

COMMON contains the two arrays P(3,200) and T(3,3,16), and the four quantities NA, NB, NP, DIA. We have seen that NA and NB are the parameters that determine the tiling. NP is the number of points in the motif, and DIA is the diameter of the absolute. The array P(3,200) contains the coordinates X, Y and the pen position, PEN, for each point of the motif. Provision is made for as many as 200 points.

The index I in the first READ statement is used to distinguish the first run (I = 1) from the second run (I = 2). The first run will print out the vertices of the initial triangle, and the second run will produce the design.

The 10 auxilliary quantities A1, A2, ..., A10 are needed to determine the vertices of the initial triangle and to form the basic transformation matrices. The DATA statement fills the T-array with zeroes. The program then proceeds to make T_1 into a unit matrix by entering 1 in each of the diagonal positions:

T(1,1,1) = 1, T(2,2,1) = 1, T(3,3,1) = 1 .

For convenience, we next make T_2, T_3 and T_4 into unit matrices by calling the EQ subroutine. Thus, CALL EQ(2,1) makes $T_2 = T_1$ and so on. We now modify T_2, T_3 and T_4 to create matrices for the reflections in the three sides of the initial triangle, as follows:

(a) Put T(2,2,2) = -1. Hence $T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. This is the matrix of a reflection in the side AC.

(b) Put T(1,1,3) = A10, T(1,3,3) = A9, T(3,1,3) = -A9, T(3,3,3) = -A10. Thus $T_3 = \begin{bmatrix} A10 & 0 & A9 \\ 0 & 1 & 0 \\ -A9 & 0 & -A10 \end{bmatrix}$, which is the matrix of a reflection in the side BC.

(c) Put T(1,1,4) = A7, T(1,2,4) = A8, T(2,1,4) = A8, T(2,2,4) = -A7. Thus $T_4 = \begin{bmatrix} A7 & A8 & 0 \\ A8 & -A7 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, which is the matrix of a reflection in the side AB.

All the other matrices we require will be formed from the matrices T_1, T_2, T_3, T_4 by the operations EQ and PR.

We now make use of the subroutine PR. The command CALL PR(5,4,2) makes $T_5 = T_4T_2$, the product of the matrices T_4 and T_2, that is, the product of a reflection AC followed by a reflection in AB. It can be shown that this sequence of two reflections results in a rotation, about A (the origin) through the angle $360^\circ/NA$. Similarly, the command CALL PR(6,3,4) makes $T_6 = T_3T_4$, which is the matrix of a rotation about B, through the angle $360^\circ/NB$. In what follows, we make use only of T_1, T_2, T_5, T_6.

It would be too large an order to give a detailed documentation of the structure and operation of HYPAT. Instead, we indicate the function of each of the 16 matrices $T_1, \ldots, T_{16}$.

T_1: Unit matrix.

T_2, T_3, T_4: Reflections in the sides AC, BC and AB of the initial triangle.

T_5: Rotation of $360^\circ/NA$ about A.

T_6: Rotation of $360^\circ/NB$ about B.

$T_7 = T_5^a$: a rotation about A.

$T_8 = T_6^b$: a rotation about B.

$T_9 = T_5^c$: a rotation about A.

$T_{10} = T_6^d$: a rotation about B.

$T_{11} = T_7T_8T_9T_{10} = T_5^aT_6^bT_5^cT_6^d$: A transformation of this type can, by proper choice of the exponents a, b, c, d be made to map the central tile into any one of the tiles in the two rings that surround the central tile.

T_{12}: A reflection in one of the diameters of the absolute, of the form $T_5^eT_2$.

T_{13}: A rotation about A, of the form T_5^e.

The two transformations T_{12} and T_{13} are used to form the central medallion, and enter into the formation of all the medallions.

T_{14}: This tranformation is either $T_{11}T_{12}$ or $T_{11}T_{13}$. Used to form the medallion in a given tile.

T_{15}, T_{16}: Matrices used for temporary storage, either in matrix multiplication, or in applying a transformation to a point of the motif.

This FORTRAN program, whose printout is given below, is adapted for use with the CDC 6600 system and the VERSATEK plotter at Indiana University. I wish to thank the many individuals at Indiana University and at IUE in Richmond, including especially my son George Alexander and Dr. Tom Osgood who have helped in the construction of this program, and who have enabled me to use the facilities.

```
      PROGRAM HYPAT(INPUT,OUTPUT,PLOT,TAPE47=PLOT)
      COMMON/LBL/P(3,200),T(3,3,16),NA,NB,NP,DIA
      DATA T/144*0./
      CALL IDENT(47)
100   FORMAT (I1)
110   FORMAT (3I3,F4.1)
120   FORMAT (3F4.3)
      READ 100,I
      READ 110,NA,NB
      PI=3.14159
      A1=COS(PI/NB)/SIN(PI/NA)
      A2=SQRT(A1*A1-1)
      A3=A2/(1+A1)
      A4=SQRT((A1*COS(PI/NA))**2-A2*A2)
      A5=A2*COS(PI/NA)/(A4+COS(PI/NA))
      A6=A2*SIN(PI/NA)/(A4+COS(PI/NA))
      A7=COS(2*PI/NA)
      A8=SIN(2*PI/NA)
      A9=2*A1*A2
      A10=1-2*A1*A1
      GO TO(130,140),I
130   FORMAT (1X,'(0,0),(',F5.4,',0),(',F5.4,',',F5.4,')')
135   WRITE (6,130)A3,A5,A6
      GO TO 210
140   READ 110,NA,NB,NP,DIA
      READ 120,P
      T(1,1,1)=1
      T(2,2,1)=1
      T(3,3,1)=1
      CALL EQ(2,1)
      CALL EQ(3,1)
      CALL EQ(4,1)
      T(2,2,2)=-1
      T(1,1,3)=A10
      T(1,3,3)=A9
      T(3,1,3)=-A9
      T(3,3,3)=-A10
      T(1,1,4)=A7
      T(1,2,4)=A8
      T(2,1,4)=A8
      T(2,2,4)=-A7
      CALL PR(5,4,2)
      CALL PR(6,3,4)
      CALL EQ(7,1)
      CALL EQ(8,1)
      CALL EQ(9,1)
      CALL EQ(10,1)
      CALL EQ(11,1)
      CALL PAT
      DO 200 I1=1,NA
      CALL EQ(8,1)
      CALL EQ(9,1)
      CALL EQ(10,1)
      N1=NB-2
      DO 190 I2=1,N1
      CALL PR(15,6,8)
      CALL EQ(8,15)
      CALL EQ(9,1)
      CALL EQ(10,1)
      CALL PAT
      N2=NA-1
      DO 180 I3=1,N2
      CALL PR(15,5,9)
      CALL EQ(9,15)
      CALL EQ(10,1)
      IF((I3.EQ.1).OR.(I2+I3.LT.4))GO TO 180
      IF(I3.EQ.NA-1) GO TO 150
      N=NB-2
      GO TO 155
150   N=NB-3
155   DO 160 I4=1,N
      CALL PR(15,6,10)
      CALL EQ(10,15)
      CALL PAT
160   CONTINUE
180   CONTINUE
190   CONTINUE
      CALL PR(15,5,7)
      CALL EQ(7,15)
200   CONTINUE
210   CALL CLOSEFF
      END


      SUBROUTINE EQ(I,J)
      COMMON/LBL/P(3,200),T(3,3,16),NA,NB,NP,DIA
      DO 500 I1=1,3
      DO 400 I2=1,3
      T(I1,I2,I)=T(I1,I2,J)
400   CONTINUE
500   CONTINUE
      RETURN
      END
      SUBROUTINE PR(I,J,K)
      COMMON/LBL/P(3,200),T(3,3,16),NA,NB,NP,DIA
      DO 800 I1=1,3
      DO 700 I2=1,3
      T(I1,I2,I)=T(I1,1,J)*T(1,I2,K)+T(I1,2,J)*T(2,I2,K)
     A+T(I1,3,J)*T(3,I2,K)
700   CONTINUE
800   CONTINUE
      RETURN
      END


      SUBROUTINE PAT
      COMMON/LBL/P(3,200),T(3,3,16),NA,NB,NP,DIA
      INTEGER PEN
      CALL EQ(12,2)
      CALL EQ(13,1)
      CALL PR(15,7,8)
      CALL PR(16,9,10)
      CALL PR(11,15,16)
      DO 1050 J=1,NA
      DO 1040 J1=1,2
      GO TO(1000,1010),J1
1000  CALL PR(14,11,12)
      GO TO 1020
1010  CALL PR(14,11,13)
1020  DO 1030 IP=1,NP
      S=P(1,IP)**2+P(2,IP)**2
      T(1,1,15)=2*P(1,IP)/(1-S)
      T(2,1,15)=2*P(2,IP)/(1-S)
      T(3,1,15)=(1+S)/(1-S)
      CALL PR(16,14,15)
      X=.5*DIA*T(1,1,16)/(1+T(3,1,16))+4.5
      Y=.5*DIA*T(2,1,16)/(1+T(3,1,16))+4.5
      IF((X.GT.0.0).AND.(Y.GT.0.0))GO TO 1025
      GO TO 1030
1025  PEN=P(3,IP)
      CALL PLOT(X,Y,PEN)
1030  CONTINUE
1040  CONTINUE
      CALL PR(15,5,13)
      CALL EQ(13,15)
      CALL PR(15,5,12)
      CALL EQ(12,15)
1050  CONTINUE
      RETURN
      END
```

FRONTIERS IN COMPUTER SIMULATIONS FOR EDUCATION AND TRAINING

Francisco J. James
President
Simtek

Recent computer developments have re-defined the academic applications of computer simulations and transformed the role of the in-house computer center. The largest impact on education stems from three (3) dynamic factors: inexpensive small computers, distributed data processing, and 16-byte-word computers. In combination, those three factors threaten the extinction of 80% of 2nd-generation (current) computer simulations and have released a 3rd-generation breed of simulations with significantly improved realism and effectiveness. The functions and objectives of the instructor, the in-house computer center, the university, and the simulation author must adapt to these latest innovations.

To a small degree, inexpensive 'small' computers have released the simpler business simulations into the hands of home computer hobbyists and white collar families. Nevertheless, their most dramatic achievement resulted from their application of other educational techniques frequently more absorbingly effective than the current (2nd) generation business simulations. Furthermore, increasing program sophistication, combined with innovations in the field of distributed data processing, have shifted the burden of processing and administration away from the instructor, and the in-house computer center onto national computer networks specializing in this line of service. The immediate implication of this shift in responsibility promotes the in-house computer center to complementary analytical support and catapults non-technical faculty into center stage of simulation education. Simultaneously, the increasing availability of 16-byte-word computers and other word-processing hardware has cleared the way for several non-business computerized simulations. While the number of business simulations published over the past year has dropped, the number of non-business computerized simulations is experiencing an exponential growth curve. The first beneficiaries of such developments will include university-level history, political science, and law programs.

This seminar will address the problems and solutions now facing each player in the simulation cycle -- from author to final user.

COMPUTER LITERACY IN THE PUBLIC SCHOOLS

Gordon Goodwin
Coordinator of Evaluation
and Computer Services
Indiana University

Joseph E. Parsons
Project Director
North Central High School

This Title III program "Using the Computer in Instruction" is designed to augment the existing mathematics and science courses at North Central High School by integrating a Computer Library Unit (C.L.U.) into present course offerings. Following this C.L.U., students and teachers would be able to use the computer as an instructional tool in most mathematics and science courses through a number of computer assisted instruction applications. Beyond teaching basic computer skills through the C.L.U., the computer can be used for 1) problem solving, 2) modeling and simulation, 3) tutorial use, computer assisted instruction (CAI), and a long range goal of 4) instructional diagnosis, computer managed instruction, (CMI).

The ten day C.L.U. was designed around the following intentions:

To provide all mathematics students with:

1. Computer awareness
2. Basic (fundamental) programming skills
3. Introduction to BASIC programming language
4. Maximum "Hands on" terminal usage
5. Computer lab facility available to students when not being used for instruction
6. A library of computer programs covering a variety of subject matter
7. Laboratory staff to assist and provide resource to both staff and students

The curricular design consists of a two-week Computer Literacy Unit which is taught in all existing mathematics courses at the high school. All mathematics and science teachers augmenting existing courses by using the computer as an instructional tool have received a variety of in-service training.

The program has a staff consisting of Project Director, and Assistant Director/Secretary, and student helpers. The Director and Assistant Director are responsible for the development, maintenance, and dissemination of Computer Lab Schedules; assist in development, production, dissemination, and control of text, test and program library materials; provide all in-service activities and materials; assist teachers and students; and generally supervise the Computer Lab.

The program is housed in the computer suite which consists of a classroom and computer lab. Entire classes as well as individual students are scheduled in and out of the computer suite throughout the school day.

A text entitled Shake Hands With a New Friend was written under the direction of the Project Director by members of the North Central High

School mathematics and science staff. The summer following the first year of the program the text was revised. There are both teachers and students editions of the text. During the second year of the program a Pre/Post-Test was created and implemented in January of 1977.

Our computer system is an NCR Century 200 and is housed in the J. Everett Light Career Center, which is adjacent to North Central High School. The computer system was being used for both vocational education and Washington Township administrative processing. The "Using the Computer in Instruction" Committee determined that the most cost efficient method to provide "hands on" experiences for the students in this program was to upgrade the existing computer hardware. Budgeted equipment monies from a Lilly Endowment Grant and Title III Grant purchased an additional 32K of memory, 2 disc drives, a multiplexer, eleven hard-wired thermal printer terminals and one portable dial up thermal printer terminal. The following year one CRT terminal and two 25-inch television monitors were purchased for the classroom in the Computer Suite.

The initial in-service program was one week (30 hours) in July 1975, and was under the direction of the Project Director and a representative from NCR. In-service programs of the same duration have been given each summer by the Project Director for any interested Washington Township staff members. During the workshop, all participants were expected to:

1. Learn how to operate the computer terminal
2. Learn the proper procedure to "sign-on" and "sign-off" the system
3. Use all of the "BASIC" program system commands
4. Write, enter and execute computer programs through the terminal using all "BASIC" program instructions
5. Write, enter and execute a program through a terminal, which would be useful to them in their classroom or office
6. As a result of the above, will have a better understanding of computer concepts and logic, and how the computer can be an effective tool to learning.

These goals were satisfactorily accomplished as evaluated by the Project Director and workshop participants.

During the first year, 2,229 students, which represent all North Central High School mathematics classes participated in the ten-day C.L.U. The second year, 1,165 students, representing only sophomore mathematics-science students, as are the 1,200 students for the 1977-78 school year. Metropolitan School District of Washington Township serves a surburban community and consists of 12 elementary schools K through 6, 3 Junior High Schools 7 through 9, one High School 10 through 12, and an area Vocational Education Center.

During the last three years our primary objective has been Computer Literacy. As a matter of information, we have, however, ventured into some other aspects of computer based education. In order to facilitate both some short and long range goals, we have begun to move the C.L.U. from the High School into the Junior Highs. With 1 portable terminal during Spring 76/77, we ran a pilot C.L.U. in each of our three Junior

High Schools, each for a six week period. A Junior High edition of our text, Shake Hands With a New Friend, was produced, teachers were in-serviced and classes were identified. The Junior High pilot is being continued this year, increasing the time factor to 11 weeks per school. To totally move the C.L.U. to the Junior High level will require additional computer hardware.

Another objective of our program for this year incorporated the computer and writing of computer programs into selected mathematics levels for the purpose of re-enforcing math skills. Math levels and teachers were identified, curriculum was written, pre/post tests designed. Treatment and control groups were identified. The data will be analyzed to determine the future of this objective.

A spin-off result of our "open" staff in-service program stimulated the interest of our Special Education staff. Enabling activities were designed for this objective and these students are interacting with programs through computer terminals to re-enforce teacher taught activities.

Our Science Department, mainly chemistry, has been identifying students who possess high level computer skills to write programs to stimulate a variety of experiments. These programs are then libraried for use by students.

PROJECT GRAMCORD: A GRAMMATICAL CONCORDANCE PACKAGE PROGRAM FOR THE GREEK NEW TESTAMENT

Paul A. Miller
Director, Project GRAMCORD
Indiana University

Every linguistics scholar recognizes the value of a simple word concordance in studying a language; a listing of a particular word in its various contexts provides valuable information for deciphering vocabulary. Even so, a word concordance remains limited by definition; it can not provide much information beyond the level of basic word studies. Unfortunately, such fields as stylistics analysis, exegesis, and even author identification rely very heavily on "higher level" phenomena---particularly grammar or syntax. Thus, a grammatical concordance could prove an immensely valuable research tool in a wide variety of language related disciplines. Such grammatical concordances are extremely rare, however, due to the herculean efforts implicit in their compilation. Those that do exist for a few selected texts tend to cover only a handful of rather basic grammatical constructions.

The repertoire of word concordances available to scholars has been greatly enlarged in recent years by the development of word concordance package programs. Accordingly, it is quite appropriate that the computer's ability to rapidly organize enormous amounts of data be applied toward the task of producing grammatical concordances.

The GRAMCORD project began at Indiana University, Bloomington in December, 1976 at the suggestion of Dr. James L. Boyer, then chairman of New Testament and Greek at Grace Theological Seminary, Winona Lake, Indiana. The specific goal was to produce a package program and suitable data files capable of concording from the Greek New Testament any type of morphologically dependent construct which a scholar might wish to define. Such a program could save hundreds of man-hours of tedious searching, while revolutionizing Biblical exegesis and translation. The more general goal was to provide a scheme by which the basic principles of the project could be applied toward any highly inflected language, i.e., any language heavily dependent on syntactic indicators. The overriding philosophy of the project demanded an optimal balance between machine effort and human intervention, i.e., letting each do what it does best, thus assuring a minimum investment of machine costs and human time.

In a quick summation, the GRAMCORD project as completed to date consists of four programs: the first three prepare basic data; the fourth does the actual concording. Program One reads the files ATEXT (the text being prepared for concording), PARSER (verb parsing information), COMWRDF (the common words of the language and their characteristics), and TERMF (the syntactically relevent word terminations). The output lists each word as it appears, its reference, and its grammatical description(s) as deduced by the program. Ambiguous or undecipherable words are flagged, and produce a list of likely descriptions. These are inspected by hand and the number of the correct

* Copyright 1978

description choice is circled. These numbers are placed on EDITF (editing information file). Program Two (EDITOR) reads the output from Program One and produces a final grammatical directory based on the input file EDITF. Program Three (ENCODER) simply encodes the user oriented grammatical directory to a form more readily digestable by the computer. Program Four (the grammatical concordance package itself) reads the encoded grammatical directory, and produces selective grammatical concordances according to the specifications of the user supplied COMMNDF (command and construct definition file). In its present form GRAMCORD concords up to twenty-five different grammatical constructions per program execution.

The GRAMCORD project began with a search for the Greek New Testament in a machine-readable form. A magnetic tape copy of A.Q. Morton's transliterated United Bible Societies text (1968) was purchased at cost through Biblical Research Associates of Wooster College, Wooster, Ohio. The revised Morton text used in the project depicts each Greek letter by an English capital letter equivalent: A for alpha, B for beta, G for gamma, D for delta, E for epsilon, Z for zeta, J for eta, Q for theta, I for iota, K for kappa, L for lambda, M for mu, N for nu, X for xi, O for omicron, P for pi, R for rho, S for sigma, T for tau, U for upsilon, F for phi, C for chi, Y for psi, and W for omega (see Exhibit A). In accordance with the older uncial manuscripts, breathings and accent diacritical marks do not appear. Manual editing inserted the book, chapter, and verse delimiters omitted from the original Morton text.

As mentioned earlier, the first of the four programs initiates preparation of the input data used by the grammatical concordance program itself (hereafter called GRAMCORD). Computerized grammar studies in other languages have often been hampered by the laborious word-by-word manual annotation of texts whereby a set of codes are placed in parentheses following each word. Program One alleviates much of this manual editing. The rather simple strategy for word analysis allows the program to write most of the GRAMCORD input data on its own while producing a data format amenable to rapid editing where necessary. To do so, the program accesses three input files. The PARSER file represents a special concession to the extreme complexity of the Greek verb system. Verb entries in PARSER occur as in their order of appearance in the text along with their reference, parsing, and root form (i.e., the first person present active indicative). The first entry in Exhibit B, for instance, identifies the word EPOIJSAMJN in Acts 1:1 as a first person singular first aorist middle indicative for the verb POIEW. The COMWRDF (COMmon WoRD File) provides much of the basic knowledge of the language by identifying the most frequently occuring words and their set or sets of grammatical descriptions. Here may be found the articles, pronouns, conjunctions, particles, and prepositions. The first entry in Exhibit C identifies the word TON as the definite article in the masculine singular accusative. Locally frequent words---i.e., words occurring very often in a particular portion of text---may be appended to the COMMRDF when the corresponding text portion is under analysis. For example, temporarily adding an entry for APOSTOLOI while processing the Acts of the Apostles allows Program One to automatically write unambiguous grammatical descriptions for all its occurences. The TERMF (TERMinations File) provides information necessary in identifying the syntactically relevant suffixes of the nouns, adjectives, and adverbs. Thus, the first entry in Exhibit D would cause Program One to list QEOFILE as a noun in the masculine singular vocative.

As Program One executes, each word from the source text is checked against information provided by the three other input files. If the incoming word matches the next entry in PARSER, the word is listed in the output with its parsing information. If not found in PARSER, the word filters down through one of ten word-length-sorted binary search trees constructed from the data in COMWRDF. Where a particular common word has more than one possible description set, the other possibilities are linearly linked back from the regular binary tree node in a sort of "third dimension". When an incoming word matches one of these ambiguous entries, the output listing for the word contains all of these possibilities. If the word is not found in the common word tree, it is processed through a hashed-linked-list structure formed from the TERMF. If the terminations match, the output includes all of the word's possible descriptions as deduced solely on the basis of its ending. In the rare case where no match has occurred after the series of three analyses, the word's output listing is flagged for manual editing. It is interesting that most of these unidentified words turn out to be Hebrew or Aramaic transliterations; in flagging them as unidentified the program recognizes their distinctive, non-Greek morphology. The optional statistical summary compiled by Program One indicates that approximately half the words in a typical corpus are fully analyzed by the program and require no further human intervention.

The ambiguous and unidentified word entries are resolved by the EDITOR (Program Two). A scholar examines the tentative grammatical description output from Program One and merely circles the number of the correct choice where an ambiguous, multiple description listing occurs (see Exhibit E). For the rare cases where a word is completely unidentified, the scholar inserts the correct grammatical description. Thus, only in the few worst cases does the scholar complete the entire analysis process himself. A keypuncher unfamiliar with Greek merely types into the EDITF (EDITing File) the number of the circled entires along with the occasional user-supplied descriptions (see Exhibit F). EDITOR reads the tentative grammatical description file written by Program One, and produces a fully edited description file based on EDITF (see Exhibit G). The resulting output not only serves as the data base for GRAMCORD, but represents a valuable reference work in itself, providing a grammatical directory to the text being studied.

ENCODER (Program Three) reads the final grammatical description file and rewrites each line in a format less bulky and more readily digestable by the final GRAMCORD program (see Exhibit H). Where advantageous, abbreviations yield to numerical codes. Words are preceded by their length, so as to allow GRAMCORD to quickly load buffer arrays without checking character-by-character for a terminating blank space. Non-essential blank lines and spaces are trimmed to conserve on peripheral storage requirements.

A GRAMCORD APPLICATION

Before reviewing the strategy and data structures used by GRAMCORD, the last of the four programs, let us look more closely at the very concept of a grammatical concordance package program and a specific application.

One distinguishing characteristic between the classical Greek of ancient Athens and the Koine Greek of New Testament times relates to the use of the genitive absolute construction. This construct is usually defined as a participle and a noun or pronoun standing together by themselves in the genitive case, where the noun or pronoun does not denote the same person or thing as the subject or object of the sentence. The latter portion of this rule is frequently relaxed in Koine Greek, as in Matthew 18:25. This type of distinction might be used in comparing various New Testament authors as to their tendencies to emulate classical Greek style. Conjoining this stylistics study with other basic tests might prove invaluable in identifying authorship, as with the anonymous Epistle to the Hebrews. A grammatical concordance package program could locate and print all the occurences of the genitive absolute for a given text along with its reference and context. The scholar could then study the entries to determine where and how often the classical constraint prevails.

Exhibit I illustrates a possible construct definition set created by the GRAMCORD user for the aforementioned study. In this case, the series of commands request a concordance for a genitive absolute based on a noun. The construct definition set consists of four parts: the individual target definitions, the "END" command, the option definitions, and the "XSTOP" command. Each target definition specifies the grammatical characteristics of one element (word) of the construct. Line 1, for example, requests a verb in the participial mood, genitive case. An asterisk in a target definition signifies a "wild card", i.e., the user is unconcerned about the grammatical characteristic denoted by that field. Thus, the six asterisks in line 1 allow the verbs sought to be in any person, number, tense, voice, participial number, and participial gender. Similarly, the genitive noun in line 2 may be of any gender or number. The "END" command terminates the target definition section. The option commands allow the user to set additional specifications; in practice, many of these option commands may be omitted, as GRAMCORD will assume its own defaults. The "CONTEXTFIELD" command in the example requires the verb and noun to occur within eight words of one another. Since no punctuation should interrupt a genitive absolute, the "PHRASING ON" command causes GRAMCORD to ignore those contexts where this condition is not realized. Grammatical agreement refers to the formal correspondance of one word with another, usually in terms of inflectional gender, number, case, or person. Each "AGREEMENT" command allows the user to demand that any two targets agree in a particular characteristic. Thus, line 7 demands that the first and second user-defined targets agree in their seventh and second fields respectively, i.e., in number. Line 8 makes a similar constraint for agreement in gender. The "XSTOP" command terminates the construct definition set and may be immediately followed by the target definitions of another construct set.

Exhibit J provides a sample GRAMCORD output for the construct set in Exhibit I. The entry claims a noun-type genitive absolute occurs in Acts, Chapter one, beginning in verse eight and ending in verse eight. The construct sought appears in context with asterisks flagging its first and last elements. By inspecting all such concordance entries, the scholar may quickly conclude that the classical constraint has been consistantly respected by the author of this portion of scripture.

GRAMCORD'S DATA STRUCTURES

All four computer programs of the GRAMCORD Project were written in the language PASCAL. Pointer-based variables simplified programming and debugging, and facilitated rapid print traversals of large data structures. The variant record feature allowed tree nodes storing words of different grammatical classes, and, therefore, vastly differing sets of information, to be stored in the same tree structure with only one node type and very little unused allocated memory. The direct use of relational operators on characters and packed character arrays aided alphabetizing, tree searching, and command processing. The self documenting orientation of the language hastened program development immensely. Recursive functions and subroutines allowed very elegant solutions for otherwise tedious batteries of tests. Admittedly, certain of these more peculiar features were employed conservatively in the final concordance program itself so as to ease possible future translation difficulties and enhance transportability.

GRAMCORD's basic strategy revolves around a series of closely related one and two dimensional arrays which the author divides into the categories of command dependent, sentence dependent, and construct dependent. The command dependent arrays are loaded prior to text analysis entirely on the basis of the user supplied COMMNDF (COMMaND File). Since GRAMCORD processes the text data one sentence at a time, the sentence dependent arrays are at least partially reinitialized with each new sentence and serve as "scratch paper" for GRAMCORD's work. The construct dependent arrays serve an even more fleeting purpose and remain stable only briefly while a particular construct match candidate is under review. At present, GRAMCORD can concord up to twenty-five different grammatical constructs per program execution. This limit will probably be increased, since the marginal cost per construct searched is relatively minimal when compared with the total cost of I/O. While a given sentence resides in memory, all of the applicable user supplied construct definition sets are matched against it.

GRAMCORD begins by digesting the COMMNDF and loading the target data and "END" commands into the two dimensional MATCHDATARRAY (see Exhibit K). At the same time, a numerical code for the target's grammatical class is loaded into its corresponding row in the one dimensional CLASSARRAY. Once the target commands are dissected in this manner, a subroutine employs the information in CLASSARRAY to load the two dimensional LINKARRAY. Each row of LINKARRAY corresponds to a grammatical class, such as noun, verb, or article; each slot in a given row is assigned a row number of an entry in MATCHDATARRAY pertaining to that same grammatical class. For instance, all of the entries in the eleventh row of LINKARRAY denote specific verb-type targets in the MATCHDATARRAY. Thus, LINKARRAY "links" each incoming word of the text to the applicable user supplied targets of the same class for further grammatical match tests. Other command dependent arrays store the specifications of the option definition commands. All of these arrays remain static throughout the execution of the program and serve as a reference framework for the concording process.

Because the sentence is the principal grammatical unit of the language, only one sentence and its descriptions need occupy memory at any given time. Each sentence dependent array provides temporary storage for information gleaned from the analysis of one sentence. GRAMCORD loads each incoming word of the text into an array of records, SENTENCEARRAY, along with its grammatical characteristics, its reference by verse number, and any punctuation that immediately follows it. Each row of the boolean HITARRAY corresponds with a target row in MATCHDATARRAY; its columns represent word positions in the sentence under review (see Exhibit L). Thus, a "true" in HITARRAY [1, 2] would indicate that a match for target number one, i.e., the first row in the MATCHDATARRAY, occured at the second word of the sentence. The zero column of HITARRAY is assigned a "true" whenever the corresponding target occurs somewhere within the sentence in general. Using this flag, only those rows actually accessed during the sentence analysis need to be checked more closely for individual matches; only rows with a "true" in the zero column need be reinitialized to all "false" when a new sentence is introduced. If all of these zero columns are "turned on" for a given construct, GRAMCORD knows that all of the user supplied targets for that construct were found in the sentence, and the program proceeds to insure that the other constraints are met. Each row in HITARRAY corresponding to an "END" command in the MATCHDATARRAY serves as a summary for the target definitions set it terminates. Each "true" in the "END" row indicates a match or "target hit" for that particular position in one of the target rows above it. When all the zero column flags in HITARRAY for a particular construct set are turned on, a GRAMCORD subroutine consults the "END" row summaries to load the OVERVIEWARRAY. Each position in this one dimensional array also represents a position in the sentence and corresponds to the particular "END" row of the construct definition set currently under examination. Where a "true" occurs in the "END" row, however, OVERVIEWARRAY stores the row number of the target hit. Thus, in Exhibit M, target one was matched in the second word of the sentence, target two in the fifth word. Consulting the order of appearance specified by the SITUATION command, GRAMCORD determines from an analysis of OVERVIEWARRAY that the target matches occur in the proper "1-2" order. While scanning the OVERVIEWARRAY, GRAMCORD loads the sentence positions of the verified hits into the construct dependent, one dimensional HITHOLDER array. Should the construct require a test for agreement, a look at HITHOLDER provides a quick link to the word pairs in SENTENCEARRAY whose fields should grammatically correspond. Thus, the HITHOLDER assigned "3-6" locates the two target hits: target one occurs as the third word in the sentence, and target two as the sixth. Another check confirms that the two matches occur within the required contextfield of eight words. The "PHRASING ON" command initiates a check for any intervening punctuation that would disqualify the construct candidate. Passing this test, GRAMCORD conducts final tests for the user specified grammatical agreement constraints. The agreement command causes GRAMCORD to consult HITHOLDER; it finds that target one and two occur in positions three and six respectively. GRAMCORD quickly compares the proper fields in the corresponding positions in SENTENCEARRAY for their equivalence, i.e., grammatical agreement. If all of these final tests yield affirmative results, GRAMCORD scans SENTENCEARRAY to print the construction's context and reference in the output listing (see Exhibit J).

Most of the aforementioned checks are conducted by recursive subroutines, since a word pattern failing to meet all of the constraints early in a sentence may still have elements which find their completion later in the sentence. Yet, since each successive test proceeds only on the basis of positive results in the previous tests, GRAMCORD wastes little time on "dead end" searches. The relatively expensive recursive tests continue only when suspicions of a located construct occurence have already proven to be extremely well founded.

POSSIBLE EXTENSIONS TO GRAMCORD

GRAMCORD in its present state provides a very versatile, research oriented package program for an almost infinite variety of grammatical construction studies. Supplementing the data bases and amending the programs to handle word functions as well as morphology and basic grammatical classifications would elevate the concordance package to study even higher level and more subtle linguistic phenomena. This extended version of GRAMCORD could be employed to locate the functional slots in an experimental slot-filler tagmemic grammar. The development and testing of such tagmemic grammars could be immensely accelerated and enhanced. Grammatical rules of Koine Greek previously unnoticed by modern scholars would surface from what has often been an overwhelming body of information. Systematic use of such a reference tool could have major implications for language theory and even psycholinguistics.

The user supplied commands for GRAMCORD II would follow the same general pattern as those in the original GRAMCORD, but with additional fields to represent the functional specifications. Accordingly, the command in Exhibit N would request a nominative case proper noun functioning as the dominant headslot tagmeme. Exhibit O demonstrates a construct set that might be formulated to concord on both a morphological and functional basis. The grammatical construction defined is the noun phrase consisting of determiner, head, determiner, and descriptor---the familiar attributive position with postposed adjective modifier.

CONCLUDING REMARKS

In closing the author would like to express his sincerest thanks to the large number of people and institutions who have made the GRAMCORD Project possible. Dr. J. Arthur Baird of Biblical Research Associates played an important part in providing initial contacts in this field and in procuring a copy of the Morton text on magnetic tape. Dr. James L. Boyer, now retired from the faculty at Grace Theological Seminary, helped organize much of the basic grammatical data used in the initial input files and continues to offer his very valuable assistance. Dr. Edgar J. Lovelady, Professor of English and Greek at Grace College, recently joined the project and is largely responsible for the planned extensions in tagmemic analysis. I am especially indebted to the Sperry and Hutchison Foundation and the William and Jenny Curry Classical Studies Scholarship Fund for their interest in this new area of research and extremely generous financial support.

EXHIBIT A

A sample of the adapted Morton text for Acts 1:1, 2:

```
$AC$/01/*01* TON MEN PRWTON LOGON EPOIJSAMJN PERI PANTWN, W QEOFILE,
WN JRXATO O IJSOUS POIEIN TE KAI DIDASKEIN *02*ACRI JS JMERAS ENTEILAMENOS
TOIS APOSTOLOIS DIA PNEUMATOS AGIOU OUS EXELEXATO ANELJMFJ.
```

EXHIBIT B

A sample of the PARSER file for Act 1:1, 2:*

```
AC0101 EPOIJSAMJN         1S1AMI     POIEW
AC0101 JRXATO             3S1AMI     ARCW
AC0101 POIEIN               PRAN     POIEW
AC0101 DIDASKEIN            PRAN     DIDASKW
AC0102 ENTEILAMENOS         1AMPNSM  ENTELLONAI
AC0102 EXELEXATO          3S1AMI     EXLEGW
AC0102 ANELJMFQJ          3S1API     ANALAMBANW
```

* It should be noted that the contents of the PARSER file need not be created from scratch, since they already exist in a number of standard reference works.

EXHIBIT C

A sample of the COMWRDF file:

```
TON ARTICLE/MSA
TJS ARTICLE/FSG
JS PRONOUN/FSG+OS
EX PREPOSI/G
PROS PREPOST/A
PROS PREPOSI/D
KAI CONJUNC/
GE PARTICLE
MJDE NEGPART/
W INTERJ/
IJSOUS NOUN/MSN
```

EXHIBIT D

A sample of the TERMF file:

```
E E/MSV       (the second "E" stands for either noun or adjective)
E E/FSV
AIS E/MPD
AIS E/FPD
EI E/XSD      (the "X" represents all genders)
WR N/XSN      (the "N" represents a noun termination)
W A/XSA       (the "A" represents an adjective termination)
```

EXHIBIT E

A sample of the tentative grammatical description file produced by Program One---the circled entries represent choices that will be preserved by the EDITOR for the final grammatical directory:

```
AC  1  1  WN            46 A PRONOUN     MPG OS
                        47 A PRONOUN     FPG OS
                        ㊽ A PRONOUN     NPG OS

AC  1  1  JRXATO        49 N VERB        3S1AMI ARCW

AC  1  1  O             ㊿ A ARTICLE     MSN
                        51 A PRONOUN     NSN OS
                        52 A PRONOUN     NPG OS

AC  1  1  IJSOUS        53 N NOUN        MSN

AC  1  1  POIEIN        54 N INFINITIVE    PRAN POIEW

AC  1  1  TE            55 N PARTICLE

AC  1  1  KAI           56 N CONJUNC

AC  1  1  DIDASKEIN     57 N INFINITIVE    PRAN DIDASKW
```

EXHIBIT F

The EDITF file: the numbers represent the circled choices in the tentative grammatical description file.

```
23
26.  PRONOUN    NPG    (In this case, the scholar "overrode" the
41                      program's own analysis.)
42
48
50
56
```

EXHIBIT G

The fully edited grammatical directory produced by the EDITOR:

```
AC  1  1 W                41 A INTERJ
AC  1  1 QEOFILE          42 A NOUN        MSV
,,,,,,,,,,,,          (commas denote a punctuation occuring in the text)
AC  1  1 WN               48 A PRONOUN     NPG OS
AC  1  1 JRXATO           49 N VERB        3S1AMI ARCW
AC  1  1 O                50 A ARTICLE     MSN
AC  1  1 IJSOUS           53 N NOUN        MSN
AC  1  1 POIEIN           54 N INFINITIVE    PRAN POIEW
AC  1  1 TE               55 N PARTICLE
AC  1  1 KAI              56 N CONJUNC
AC  1  1 DIDASKEIN        57 N INFINITIVE    PRAN DIDASKW
AC  1  2 ACRI             58 A PREPOSI     G
```

EXHIBIT H

ENCODER transforms the grammatical directory into a form which serves as the data base for GRAMCORD:

```
N 1W 6
N 7QEOFILE1OMSV
P,
N 2 WN 1NPGOS
N 6JRXATO11 3S2MI     4ARCW
N 1O 2MSN
N 6IJSOUS1OMSN
N 6POIEIN11   OAN     5POIEW
N 2TE 5
N 3KAI 4
N 9DIDASKEIN11   OAN        7DIDASKW
V 2
N 4ACRI 3G
```

EXHIBIT I

A construct definition set which might be used for concording a noun-type genitive absolute:

```
VERB/****PG**.
NOUN/**G.
END.
SITUATION;02/01,02.
CONTEXTFIELD;08.
PHRASING ON.
AGREEMENT:01,07/02,02.
AGREEMENT:01,08/02,01.
XSTOP.
```

EXHIBIT J

An entry in GRAMCORD's output resulting from the above commands:

```
AC  1  8- 8 LJMYESQE *EPELQONTOS TOU AGIOU  *PNEUMATOS EF UMAS, KAI ESESQE
            MOU MARTURES EN TE IEROUSALJM KAI SAMAREIA KAI EWS TJS GJS.
```

EXHIBIT K

Portions of the CLASSARRAY and MATCHDATARRAY corresponding to the target definitions in Exhibit I:

	CLASSARRAY	MATCHDATARRAY
(row 1)	11	****PG**.
(row 2)	1	**G.
(row 3)	0 [denotes "END" command]	

EXHIBIT L

A portion of the HITARRAY corresponding to the above arrays when Acts 1:8 is under analysis:

```
column number: 012456789 etc.

    (row 1)     1010000000 ...
    (row 2)     1000010000 ...
    (row 3)     1010010000 ... [The "END" row]
```

EXHIBIT M

The OVERVIEWARRAY corresponding to the "END" row in Exhibit L---each number denotes a target definition number:

```
010020000 ...
```

EXHIBIT N

NOUN/**NPH.   (nom., proper noun, occupying headslot)

EXHIBIT O

```
ARTICLE/***D.
NOUN/****H.
ADJECTIVE/***S.    ("S" = description)
END.
SITUATION:04/01,02,01,03.
CONTEXTFIELD:08.
PHRASING ON.
AGREEMENT:01,01/02,01.
AGREEMENT:01,02/02,02.
AGREEMENT:01,03/02,03.
AGREEMENT:02,01/03,01.
AGREEMENT:02,02/03,02.
AGREEMENT:02,03/03,03.
XSTOP.
```

A resulting output:

```
AC  8 29-29 PARJGGELLEN GAR *TW PNEUMATI TW *AKAQARTW EXELQKIN APO TOU
            ANQRWPOU.
```

APPENDIX

The following examples of user created construct definitions demonstrate other possible applications for the GRAMCORD package.

* AKOUW with genitive pronoun

```
     VERB/********AKOUW.    (requests any verb form of AKOUW)
     PRONOUN/**G.
     END.
     CONTEXTFIELD:06.
     PHRASING ON.
     XSTOP.
```

A sample output:

```
AC  1  4- 4 ALLA PERIMENEIN TJN EPAGGELIAN TOU PATROS JN *JKOUSATE *MOU;
AC 17 32-32 OI DE EIPAN: *AKOUSOMEQA *SOU PERI TOUTOU KAI PALIN.
```

*A group of construct definition sets submitted together:

- nom.,pl.,acc. article and a prepositional phrase in the genitive with an article-noun object
- TE KAI ("both ... and") relating two active infinitives
- uses of POIEW in the middle voice

```
    ARTICLE/NPA.
    PREPOSI/G.
    ARTICLE/**G.
    NOUN/**G.
    END.
    CONTEXTFIELD:06.
    AGREEMENT:03,01/04,01.
    AGREEMENT:03,02/04,02.
    XSTOP.
    VERB/***AN***.
    PARTICLE/.
    CONJUNC/.
    END.
    SITUATION:04/01,02,03,01.
    CONTEXTFIELD:06.
    PHRASING ON.
    XSTOP.
    VERB/***M****POIEW.
    END.
    XSTOP.
```

```
(C 3) AC  1  1- 1 TON MEN PRWTON LOGON *EPOIJSAMJN PERI PANTWN, W QEOFILE,
(C 2) AC  1  1- 1 QEOFILE, WN JRXATO O IJSOUS *POIEIN TE KAI *DIDASKEIN
                  ACRI JS JMERAS ENTEILAMENOS
(C 1) AC  1  3- 3 OPTANOMENOS AUTOIS KAI LEGWN *TA PERI TJS *BASILEIAS
                  TOU QEOU;
```

[The number in parentheses denotes the construct concorded.]

* Debate over the modern charismatic movement centers a great deal of commentary on the meaning of "EN" in I Corinthians 12:13. Some would translate the preposition as "in", others as "by"; the former would denote instrumentality, the latter personal agency. One exegete claims that "EN" is never used for the personal agent with a passive voice. While GRAMCORD cannot provide a final definitive answer to this problem, it could conduct a quick search for all of the occurences of "EN" with a passive verb. With such a list assembled in a neat and concise form, the scholar can more conveniently make his own interpretive determinations.

LIST OF PARTICIPANTS

ABELES, HAROLD F., Associate Professor of Music Education, Indiana University School of Music; B.S., University of Connecticut, 1966; M.A., 1968; Ph.D. University of Maryland, 1971.

ALEXANDER, GEORGE A., Systems Coordinator, Indiana Daily Student; B.A., Earlham College, 1973; M.A., Indiana University, 1976.

ALEXANDER, HOWARD, Professor of Mathematics Emeritus, Earlham College; B.A., University of Toronto, 1933; M.A., 1934; Ph.D., Princeton University, 1939.

ANDREE, ROBERT G., Director of Computing Services, Indiana University Northwest; B.S., United States Military Academy, 1956; M.B.A, University of Georgia, 1972.

BARRY, RONALD A., Department of Statistics and Computer Science, West Virginia University; B.S., Ohio University.

CHILDERS, MARIE E., Research Associate, Graduate School, Indiana University; B.A., Southern Illinois University, 1962; M.S. in Ed., 1964; Ed.D., Indiana University, 1968.

DUFALA, MARGARET M., Department of Political Science, The University of Akron.

DUSTMAN, JOHN H., Associate Professor of Zoology, Adjunct Associate Professor of Physiology, Northwest Center for Medical Education (Indiana University School of Medicine); B.S., Canisius College, 1961; Ph.D., Indiana University, 1966.

GOODWIN, GORDON, Coordinator of Evaluation and Computer Services, Indiana University Developmental Training Center; B.A., Wabash College, 1972; M.S., Indiana University, 1975.

GROOMER, S. MICHAEL, Assistant Professor of Accounting, Indiana University School of Business; B.S., University of Evansville, 1966; M.A., University of Missouri, 1967; Ph.D., 1975; C.P.A., 1971; C.D.P., 1977.

JAMES, FRANCISCO J., President, Simtek, Graduate of Harvard Business School.

JONES, ERNEST L., Director of Data Systems and Services, Indiana University; B.A., Marshall University, 1952; M.A., 1952.

KENNEDY, MADELEINE M., Research Consultant, Wrubel Computing Center, Indiana University; B.A., Indiana University, 1975; M.S., 1976.

LAMBERT, H. DAVID, Applications Programmer, Consultant, Wrubel Computing Center, Indiana University; B.A., University of West Virgina, 1971; M.A., University of West Virginia, 1973.

LANE, MALCOLM G., Associate Professor and Associate Chairman of Computer Science, West Virginia University; B.S., Davidson College, 1965; Ph.D., Duke University, 1971.

LEE, PAUL S.T., Senior Research Associate, Center for Applied Urban Research, The University of Nebraska at Omaha; M.A., National Chung-hsing University, 1962; Ph.D., Washington State, 1973.

MARQUETTE, JESSE F., Associate Professor, Department of Political Science, The University of Akron; Ph.D.

MILLER, PAUL A., Part-Time Consultant and Instructor in Programming Languages, Wrubel Computing Center, Indiana University; A.B., Indiana University.

NERHEIM, ROSALEE J., Consultant, Wrubel Computing Center, Associate Instructor in Music Theory, Indiana University; B.M., Indiana University, 1976.

PARSONS, JOSEPH E., Project Director of Computer Based Instruction at North Central High School.

PEPPERDINE, WARREN H., Professor of Speech and Theatre, Indiana University at South Bend; A.B., University of Washington, 1955; A.M., 1956; Ph.D., University of Minnesota, 1965.

ROBERTS, STEPHEN D., Director, Health Systems Research, Indiana University Medical Center; B.S., Purdue University, 1965; M.S., 1966; Ph.D., 1968.

SCHIER, JAMES S., Systems Analyst, Indiana University Medical Center; B.S., Purdue University, 1976.

SHREINER, SCOTT C., Assistant Professor of Sociology, Indiana University-Purdue University at Fort Wayne; B.S., Pennsylvania State University, 1970; M.A., 1973; Ph.D., University of Massachusetts, 1977.

STENTZ, MICHAEL C., Regional Campus Coordinator, Applications Programmer, Research Consultant, Wrubel Computing Center, Indiana University; B.A., Southern Methodist University, 1966; M.S., University of Houston, 1970; Ph.D., Indiana University, 1979.

SULANKE, THOM, Programmer/Analyst, Indiana University Medical Center; A.B., Earlham College, 1968; M.A., Indiana University, 1975, M.S., 1976.

WAKEFIELD, GORDON D., Coordinator for Academic Computer Services, Indiana University-Purdue University Fort Wayne; B.S., University of Utah, 1962; M.S., Brigham Young, 1965.

WIDES, JEFFREY W., Faculty Applications Consultant, Southern Illinois University at Edwardsville; A.B., Washington University, 1964; M.A., U.C.L.A., 1967; Ph.D., U.C.L.A., 1972.